공학석사학위논문

오디오 쿼리 기반 음원 분리 연구

AUDIO QUERY-BASED MUSIC SOURCE SEPARATION

2020년 8월

서울대학교 융합과학기술대학원

융합과학부 디지털정보융합전공

이 지 환

공학석사학위논문

오디오 쿼리 기반 음원 분리 연구

AUDIO QUERY-BASED MUSIC SOURCE SEPARATION

2020년 8월

서울대학교 융합과학기술대학원

융합과학부 디지털정보융합전공

이 지 환

오디오 쿼리 기반 음원 분리 연구

AUDIO QUERY-BASED MUSIC SOURCE SEPARATION

지도교수 이 교 구

이 논문을 공학석사 학위논문으로 제출함

2020년 8월

서울대학교 융합과학기술대학원

융합과학부 디지털정보융합전공

이 지 환

이지환의 공학석사 학위 논문을 인준함

2020년 8월

위 원 장: _____

부위원장: _____

위　　원: _____

요 약

최근 몇 년 동안, 음악 음원 분리는 음악 정보 검색 분야에서 가장 활발하게 연구가 이루어진 분야 중 하나이다. 또한 딥 러닝의 발전으로 인해 음악 음원 분리 성능은 큰 폭으로 향상했다. 그러나 대부분의 이전 연구들은 단일 악기 또는 보컬, 드럼, 베이스와 같은 제한된 수의 음원을 분리하는데 그쳤으며, 확장성에 대한 연구는 많이 이루어지지 않았다.

본 연구에서는 오디오 쿼리 기반 음원 분리를 위해 목표 신호의 수 또는 종류에 관계없이 쿼리 신호로부터 소스의 정보를 인코딩할 수 있는 네트워크를 제안한다. 제안된 기법은 쿼리 인코딩 네트워크와 음원 분리 네트워크로 구성된다. 오디오 쿼리와 합성 음원이 주어지면 쿼리 인코딩 네트워크는 쿼리를 잠재 공간으로 인코딩하고, 음원 분리 네트워크는 잠재 벡터에 의해 컨디셔닝된 마스크를 출력하며, 이 마스크는 합성 음원에 곱해져 음원을 분리한다. 또한 음원 분리 네트워크는 학습 샘플에서 얻어진 잠재 벡터를 사용하여 오디오 쿼리가 주어지지 않은 환경에서도 동작할 수 있다.

제안한 기법의 평가를 위해 MUSDB18과 Slakh을 이용하며, 실험 결과는 제안된 기법이 단일 네트워크로 여러 소스를 분리할 수 있음을 보인다. 또한, 잠재 공간에 대한 분석을 통해 제안된 기법이 잠재 벡터의 보간을 통해 연속적인 출력을 생성할 수 있음을 보인다.

주요어: 오디오 쿼리, 음원 분리
학 번: 2018-22176

차 례

표 차례

그림 차례

제 1 장 서론

1.1 연구 배경

음원 분리(audio source separation)은 여러 신호가 포함되어 있는 음원에서 특정 신호만을 분리하는 분야이다. 음원 분리의 세부 분야 중 본 연구에서 다루는 음악 음원 분리(music source separation)는 여러 악기의 신호가 혼합된 음원에서 특정 악기의 신호를 분리하는 것으로 음악 정보 검색에서 가장 활발한 연구가 이루어지는 분야 중의 하나다. 음악에서 분리된 특정 악기의 소리는 다양하게 활용된다. 특정 곡에서의 일부분을 가져와 새로운 곡의 작곡에 사용하는 샘플링, 음원에서의 악기 간 볼륨 조절, 모노 또는 스테레오의 음원을 더 높은 채널 수의 음원으로 바꾸는 작업 등 음악을 이루는 악기들의 단일 음원을 얻을 수 없는 환경에서 음악을 편집하는데 폭 넓게 사용되고 있다.

또한, 음원 분리는 음악을 분석하는 연구에서도 전처리 과정에서 사용되어 성능 향상을 이끌어낼 수 있다. 비트 트래킹, 자동 채보, 가사 정렬과 같은 작업들은 음악

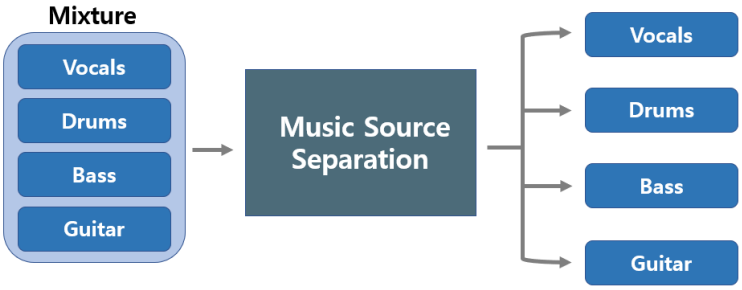


그림 1.1: 음악 음원 분리 개요

그림 1.2: 악기 범주 및 종류

을 이루고 있는 여러 악기들 중 특정 악기의 신호를 분석하는 방식으로 이루어진다. 따라서 분석에 필요한 악기의 신호만을 분리한 뒤 위의 작업을 수행하는 네트워크에 입력으로 넣어줌으로써 성능을 높이는 것이 가능하다.

이처럼 음원 분리 기술은 높은 활용 가치로 인해 많은 연구가 이루어졌다. 특히 딥러닝의 발달 이후 음원 분리에도 이를 적용한 다양한 연구가 제안되었다. 딥러닝 기술의 발전 덕에 음원 분리의 성능은 높아졌지만 그에 반해 분리 가능한 악기의 종류나 확장성에 관한 연구는 많이 이루어지지 않았다. 현재, 음원 분리를 위해 주로 사용되는 데이터셋[10, 14]에는 4개의 클래스가 포함되어 있다. 분리해야 하는 악기의 종류가 많지 않기 때문에 악기별로 분리 네트워크를 따로 학습하는 방식의 연

구[23, 24, 25]가 제안되었지만 분리해야할 악기의 종류가 늘어나는 경우 이러한 방식은 효율적이지 못하다. 이와 관련하여 단일 네트워크로 원-핫(one-hot) 벡터를 컨디셔닝하는 방법으로 고정된 수의 악기를 분류하는 방법이 제안되었다[17, 20, 31].

원-핫 벡터를 사용하는 방법은 가장 직관적인 접근이지만 넓게 정의된 악기 정보만 있는 상황에서 이상치(outlier)를 대처하기에는 부적절하다는 한계점이 존재한다. 여기서 이상치란 분리하고자 하는 악기가 넓게 정의된 한 종류의 범주에 포함되지만 그 악기의 일반적 특성과는 거리가 있는 경우를 뜻한다.

예를 들어, 주로 메탈 장르의 음악에서 주로 들을 수 있는 뭉개는 듯한 소리의 '브루털 창법을 이용하는 목소리' 또는 '어쿠스틱 기타'의 신호를 분리하려는 상황을 가정할 수 있다. 이러한 경우에 광범위하게 정의된 '보컬(vocals)', '기타(guitar)'를 사용하는 대신 더 세분화된 '브루털 창법 보컬', '어쿠스틱 기타'와 같은 정보를 사용할 수 있다면 음원 분리 성능의 향상을 기대할 수 있다.

이를 해결하기 위한 가장 단순한 방법으로는 악기 온톨로지에 따라 일일이 악기의 범주를 세분화한 뒤 이에 따라 음원을 분리하도록 학습하는 것이다. 하지만 사람이 각 음원에 수동으로 레이블링을 하는 방법은 다음과 같은 한계점이 존재한다. 첫째로, 비용이 많이 든다. 둘째로 데이터의 총량이 한정된 상황에서 범주를 더 세분화하여 분류하는 것은 각 클래스마다 샘플의 수가 적어지게 만드는 것이므로 성능 하락이 발생할 수 있다. 더불어 악기는 그림 1.2[1]와 같이 종류가 매우 다양하기 때문에 세분화하는 경우 어느 수준까지 세분화할 것인지 정하는 것도 쉽지 않다. 마지막으로 새로운 이상치가 나타나는 경우 이에 대한 확장성이 부족하다.

[1] 음악사 연구회, http://musichistory.or.kr/article/naver

1.2 연구 목표

앞서 서술한 문제들을 해결하기 위해, 본 연구에서는 오디오 쿼리 기반 음원 분리 프레임워크를 제안한다. 오디오 쿼리 기반 음원 분리란 사용자가 분리하고자 하는 신호를 지정하기 위한 방법으로 목표 신호와 유사한 오디오 쿼리를 입력으로 사용하는 방식을 의미한다.

따라서 제안한 기법은 음원 분리 네트워크만을 사용하는 전통적인 음원 분리 연구와 달리 쿼리 인코딩 네트워크를 추가한 구조를 갖고있다. 쿼리 인코딩 네트워크가 다양한 오디오 쿼리를 잠재 벡터로 직접 압축한 다음 인코딩된 잠재 벡터(latent vector)를 음원 분리 네트워크로 전달하여 합성 음원에서 특성이 쿼리와 유사한 소스(source)를 분리하는 방식으로 동작한다. 위 구조에서 음원 분리 네트워크는 잠재 벡터를 보조 입력으로 받기 때문에 학습 데이터셋의 샘플들을 미리 인코딩한 벡터를 저장해두고 오디오 쿼리가 없는 환경에서 사용하는 방식으로도 동작이 가능하다. 제안한 프레임워크는 쿼리 인코딩 네트워크가 처음 보는 목소리 또는 악기 소리를 연속적인 잠재 공간으로 인코딩하며 같은 클래스에 포함된 오디오라도 서로 다른 점으로 맵핑하기 때문에 확장성이 있다.

본 연구에서 제안한 프레임워크는 다음과 같은 장점을 갖는다. 첫째, 단일 네트워크로 다양한 소스를 분리할 수 있다. 둘째, 사용자가 분리하려는 신호와 유사한 것으로 간주하는 오디오 샘플을 직접 쿼리로 전달하여 인코딩할 수 있기 때문에 음원 내 분리하려는 신호의 특성이 해당 악기의 일반적인 특성과 다를 때 분리 성능의 증가를 기대할 수 있다. 셋째, 연속적인 잠재 공간에 존재하는 벡터를 조절하여 분리 네트워크의 출력의 제어가 가능하다.

본 연구에서는 제안된 프레임워크의 유용성을 입증하기 위해 MUSDB18 데이터셋[14]과 Slakh 데이터셋[38]을 이용하여 다양한 실험을 진행한다. 주어진 쿼리에

따라 특정 신호를 분리해내는 것을 보이고, 실험을 통해 분리 네트워크의 출력이 잠재 벡터에 의해 제어되며 잠재 벡터를 보간(interpolation)하여 출력이 신호 수준에서 원활한 전환이 가능한 것을 보인다.

또한 일반적인 특성과 다른 신호를 분리할 때 분리에 사용되는 잠재 벡터의 정제를 자동화할 수 있는 방법을 소개한다. 더불어 정량적 평가를 통해 오디오 쿼리 기반 음원 분리의 성능 평가에 대한 기준을 제시하고 기존 음원 분리 연구들과의 비교를 통해 제안한 프레임 워크가 전통적인 음원 분리 네트워크로써도 동작할 수 있음을 보인다.

제 2 장 배경 이론 및 관련 연구

2.1 배경 이론

본 절에서는 음원 분리 연구의 개괄적인 내용과 함께 본 연구에서 사용한 쿼리 인코딩 네트워크의 근간이 되는 variational autoencoder에 관련된 이론적 배경에 대한 설명을 서술한다.

2.1.1 음원 분리

음악 음원 분리 (music source separation)은 여러 신호의 합성 음원에서 특정 신호를 분리해내는 기법을 의미한다. 여러 종류의 음원이 동시에 존재할 때, 관측된 시간 영역 신호의 파형 $x(t)$은 개별 음원 $s_n(t)$의 합으로 나타낼 수 있다.

$$x(t) = \sum_{n=1}^{N} s_n(t) \tag{2.1}$$

N은 전체 개별 소스의 수를 나타내며 따라서 음원 분리는 $x(t)$에서 특정 신호 $s_n(t)$를 분리하는 것이다. 이에 관련하여 초기에는 공간적 위치를 고려하고 소스의 통계적 독립성에 기초하여 혼합 신호의 분해 매트릭스를 추정하는 독립 성분 분석 (Independent Component Analysis, ICA)를 이용했다. 그러나 이러한 방식은 합성 음원에 포함된 소스와 동일한 수의 채널을 가진 음원을 요구하기 때문에 한계점을 가졌다.

이후 음원의 크기 스펙트로그램(magnitude spectrogram)에 비음수 행렬 분해

(Non-negative matrix factorization, NMF)을 적용한 연구들이 대두되었다. 비음수 행렬 분해는 스펙트로그램을 각 소스의 스펙트럴 특성을 담은 기저 벡터와 그에 대한 각 시간 활성도(time activation)의 곱으로 나타내는 과정으로 동작한다. 하지만 일반적으로 목소리와 같이 복잡도가 높은 신호의 경우 분리 성능이 높지 못한 모습을 보였다.

딥러닝의 발전 이후 음원 분리 성능은 비약적으로 높아졌다. 합성 음원과 합성 음원을 이루는 개별 소스의 음원이 주어진 데이터셋을 바탕으로 지도 학습(supervised learning) 기반의 인공신경망을 학습하는 방식은 많은 관심을 끌었다. 딥러닝을 이용한 음원 분리의 경우 일반적으로 다음의 과정과 같다. 신호를 단시간 푸리에 변환(short-time-Fourier-transform, STFT)를 이용해 스펙트로그램으로 변환한다. 이후 위상 정보는 사용하지 않고 크기 스펙트로그램을 딥러닝 네트워크의 입력으로 사용한다. 네트워크는 입력에서 분리하려는 신호만을 남기기 위해 시간-주파수의 마스크를 생성하고 마스크는 이진 마스크(0 또는 1의 값만을 가지는 마스크) 또는 소프트 마스크(0에서 1 사이의 값을 갖는 마스크)의 형태로 나뉜다. 출력된 마스크는 입력에 곱해진 뒤 합성 음원의 위상 정보를 이용하여 역단시간 푸리에 변환(inverse-STFT)를 거쳐 음원으로 복원한다. 최근에는 이러한 방식이 위상을 고려하지 않는다는 한계점을 지적하며, 시간 도메인에서 직접 음원 분리를 수행하거나[21] 분리 과정에서 위상 정보까지 고려하는 방법도 제안되었다[18].

2.1.2 Variational Autoencoder

Variational Autoencoder(VAE)[7]는 생성모델로써 데이터의 확률분포$p(x)$를 학습하고 데이터를 생성하는 것이 목적이다. 그림 2.1는 인코더와 디코더로 구성되는 VAE의 구조를 나타낸 것이다. 디코더는 잠재 변수 z를 바탕으로생성하는데, 데이터와 유사하게 생성되기 위해서는 z가 이상적인 확률분포인 $p(z|x)$로부터 샘플링되어

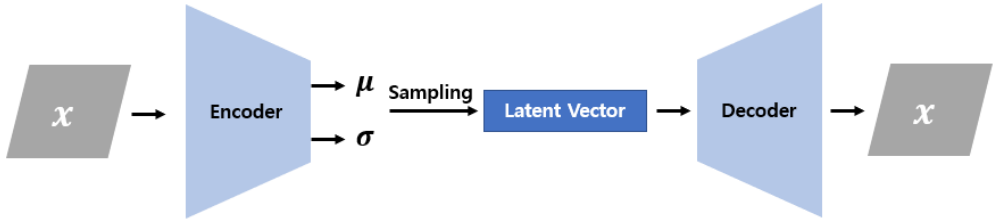


그림 2.1: Variational Autoencoder의 구조

야 한다. 하지만 $p(z|x)$에 대해 알 수 없으므로 인코더를 이용해 $q_\phi(z|x)$를 $p(z|x)$에 근사시킨다. 이처럼 확률분포를 추정하기 위해 가우시안 분포 등을 가정하고 파라미터를 조정하여 근사시키는 것을 variational inference라 한다. $p(x)$를 구하기 위해 수식을 전개하면 다음과 같다.

$$\log(p(x)) = \log\left(\int p(x,z)dz\right) = \log\left(\int p(x|z)p(z)dz\right) \tag{2.2}$$

식 2.2에 Jensen's Inequality를 적용하면 다음과 같은 부등식이 성립한다.

$$\log(p(x)) = \log\left(\int p(x|z)p(z)dz\right) \geq \int \log(p(x|z))p(z)dz \tag{2.3}$$

식 2.3의 우변에 $q_\phi(z|x)$ 항을 넣고 전개시키면 다음과 같다.

$$\log(p(x)) \geq \int \log\left(p(x|z)\frac{p(z)}{q_\phi(z|x)}\right) q_\phi(z|x)dz \tag{2.4}$$

$$= \int \log(p(x|z))q_\phi(z|x)dz - \int \log\left(\frac{q_\phi(z|x)}{p(z)}\right) q_\phi(z|x)dz \tag{2.5}$$

$$= \mathbb{E}_{q_\phi(z|x)}[\log(p(x|z))] - KL\left(q_\phi(z|x)||p(z)\right) \tag{2.6}$$

식 2.6를 $ELBO$(Evidence lower bound)라 부르며 이때 $ELBO(\phi)$를 최대화하는 ϕ 값을 찾으면 $\log(p(x))$와 같게 된다. 여기서 $\log(p(x))$를 $\int q_\phi(z|x)dz = 1$을 이용해 $\log(p(x))$을 다음과 같이 바꿀 수 있다.

$$\begin{aligned}
\log(p(x)) &= \int \log(p(x)) q_\phi(z|x) dz && (2.7)\\
&= \int \log\left(\frac{p(x,z)}{p(z|x)}\right) q_\phi(z|x) dz && (2.8)\\
&= \int \log\left(\frac{p(x,z)}{q_\phi(z|x)} \cdot \frac{q_\phi(z|x)}{p(z|x)}\right) q_\phi(z|x) dz && (2.9)\\
&= \int \log\left(\frac{p(x,z)}{q_\phi(z|x)}\right) q_\phi(z|x) dz + \int \log\left(\frac{q_\phi(z|x)}{p(z|x)}\right) q_\phi(z|x) dz && (2.10)\\
&= ELBO(\phi) + KL\left(q_\phi(z|x) \| p(z|x)\right) && (2.11)
\end{aligned}$$

식 2.10의 왼쪽 항은 식 2.6과 같으며 $p(z|x)$에 대해서 모르기 때문에 $ELBO$를 최대화하는 ϕ를 찾는다. 식 2.6의 왼쪽 항은 z를 입력으로 받아 x를 복원하는 디코더의 역할과 같다. 따라서 θ를 이용해 이를 최대화한다. 최종 손실 함수는 다음과 같이 정의 될 수 있다.

$$\mathcal{L}_{(\theta,\phi;x^i)} = -\mathbb{E}_{q_\phi(z|x^i)}\left[\log\left(p_\theta\left(x^i|z\right)\right)\right] + KL\left(q_\phi\left(z|x^i\right) \| p(z)\right) \quad (2.12)$$

식 2.12에서 좌변의 왼쪽 항은 복원(reconstruction) loss를 의미하며 몬테-카를로 방법을 이용한다. 일반적으로 샘플링 횟수는 1로 사용한다. 이때 인코더에서 출력되는 것은 $q_\phi(z|x)$의 모수 μ, σ인데 이를 이용해 샘플링($z^{i,l} \sim N\left(\mu_i, \sigma_i^2\right)$)할 경우 샘플링은 미분이 불가능하기 때문에 역전파(back propagation)가 불가능해지는 문제가 발생한다. 이 문제를 해결하기 위해 $\epsilon \sim \mathcal{N}(0, \mathcal{I})$을 샘플링하여 식 2.13과 같이

확률적 특성을 보존하면서도 미분이 가능하게 바꾸는 방법을 사용한다. 이 방법을 reparameterization trick이라 한다.

$$z^{i,l} = \mu_i + \sigma_i^2 \odot \epsilon \tag{2.13}$$

2.2 관련 연구

이번 절에서는 사전에 진행된 음악 음원 분리 연구들 중 본 연구와 관련이 있는 연구들과 앞선 장의 연구배경에서 언급한 음원 분리 연구의 확장성을 높이기 위한 방법을 제안한 다른 연구들에 대해서 살펴본다. 또한 음악 음원 분리를 다루지는 않았지만 본 연구와 관련이 있는 연구들에 대해서도 서술한다.

2.2.1 음원 분리 연구

본 절에서는 음악 음원 분리를 다룬 사전 연구들에 대해 설명한다. 우선 음악 음원 분리 관련 연구 중 부가적인 정보를 이용하여 성능을 이끌어낸 논문들을 살펴본다. 또한, 음원 분리 연구들 중 단일 네트워크를 이용해 여러 종류의 소스를 분리하는 방법을 다룬 연구들을 차례대로 살펴보며 그들의 특징과 한계점을 다룬다.

전통적인 음원 분리 연구들은 비음수 행렬 분해을 이용한 방식이 주를 이루었다. 그 중 성능을 높이기 위해 분리하려는 음원과 관련된 부가적인 정보를 사용한 인폼드 음원 분리(informed source separation) 연구들이 있었다. *Ewert*는 행렬 분해 과정 중에 악보를 이용하고 template 벡터와 활성화 벡터에 제한을 두어 성능를 높였다[1]. 또한, NMF를 기반으로하며 사용자에게 음원의 오디오의 시간-주파수 표현 형태인 스펙트로그램을 보여주고 사용자에게 분리를 원하는 소스의 기본 주파수를

고르게하고 선택된 기본 주파수를 바탕으로 음원 분리를 수행하는 시스템을 제안한 연구도 있었다[4].

반면, 단일 네트워크로 여러 종류의 악기를 분리하는 방법들 중의 하나로 부가적인 정보를 이용하는 대신 음원 분리 네트워크의 출력 채널 수를 늘리는 방법으로 접근한 연구가 있다[13]. 해당 연구에서는 hourglass 네트워크[12]를 사용하여 음원을 입력으로 받아 분리할 악기 별로 출력 채널을 따로 두어 마스크를 생성하는 방식을 제안했다. 또한 hourglass 네트워크를 1개만 사용하는 것이 아니라 최대 4개까지 연결하여 분리하고자 하는 소스의 작은 성분까지 집중하는 효과를 이끌어내어 분리 성능을 끌어올린 것이 특징이다. 제안한 방식은 DSD100 데이터셋[10]을 이용해 다른 알고리즘과의 평가를 진행했는데, 한 네트워크로 여러 음원을 분리하면서도 몇 악기에 대해서는 최고 수준(state-of-the-art)에 준하는 성능을 보이기도 했다. 그러나 출력에 모든 악기에 대한 마스크가 생성되고 이를 바탕으로 학습하다보니 상대적으로 분리가 쉬운 특정 악기에 대해서만 쉽게 학습하는 경향을 보이기도 했다. 게다가 이러한 방식은 분리 가능한 소스가 정해져있기 때문에 유연성이 떨어진다는 한계점이 존재한다.

*Slizovskaia*는 원-핫 벡터를 이용하여 컨디셔닝하는 방법을 제안했다[20]. Wave-U-net[21]에 컨디셔닝 방법을 접목시킨 것이 특징인데 네트워크의 bottleneck 부분에 원-핫 벡터를 컨디셔닝하는 방식을 제안했다. 컨디셔닝 방식은 원-핫 벡터를 곱하는 방식으로하여 분리하려는 소스의 정보만을 남기고 이외의 정보가 없어지도록 하는 아이디어에 착안했다. 원-핫 벡터를 이용하는 방식은 앞서 언급한 연구의 방식보다 유연하지만 서론에서 서술했듯이 이상치에 대해 대응하기 어렵고 잠재 공간을 고려하지 않기 때문에 네트워크의 출력의 조절이 어렵다.

*Meseguer-Brocal*도 앞서 서술한 연구와 비슷한 방식의 방법을 제안했다[31]. 크기 스펙트로그램(magnitude spectrogram)을 입력으로 하는 U-net[16]을 기반으로 분

리하고자 하는 소스를 원-핫 벡터를 이용해 지정하고 컨디셔닝하는 방식을 보여주었다. *Slizovskaia*의 연구와 다른 차이점으로는 bottleneck 부분이 아닌 U-net의 인코더 부분에 요소간 선형 변환(Feature-wise Linear Modulation, FiLM)[32]을 이용했다는 점이 있다. 인코더 부분에 사용한 이유로는 입력으로 주어진 혼합 음원에서 요소간 선형 변환을 이용한 컨디셔닝을 통해 분리하고자 하는 성분만 남기고 디코더 부분은 인코딩된 정보에 맞추어 동작하는 일반적인 방식으로 학습하기 위함이라고 주장한다. 또한 실험을 통해 이 실험에서 제안하는 하나의 네트워크로 4 종류의 소스를 분리하는 방식의 성능과 각 소스에 대해 분리하는 개별 네트워크를 훈련했을 때의 성능이 큰 차이가 없다는 것을 보였다. 하지만 이 연구도 원-핫 벡터를 사용하기 때문에 본 연구에서 제안하는 방식보다 확장성이 떨어진다고 볼 수 있다.

*Seetharaman*은 딥 클러스터링[17] 방식을 이용하여 각 시간-주파수 성분을 고차원으로 맵핑하는 방식을 제안했다[2]. 이후 별도의 네트워크를 이용해 가우시안 혼합 모델의 파라미터를 예측하여 각 소스별 마스크를 생성하는 방식을 사용했다. 하지만 위 방식은 소리를 직접 잠재 공간으로 압축하는 방식이 아니라는 단점이 있다.

다음으로 음악 음원 분리는 아니지만 화자 분리에서 비슷한 방식으로 접근한 연구들도 있었다. *Wang*은 다화자 음원에서 특정 화자의 음성을 분리하기 위해 음성을 앵커 벡터로 직접 압축하는 네트워크를 제안했다[26]. 압축한 앵커 벡터를 이용하여 시간-주파수 공간 상의 임베딩이 같은 화자의 앵커 벡터와 클러스터링 될 수 있음을 보였다. 또한 같은 분야에서 화자 임베딩 네트워크와 음성 분리 네트워크를 이용하여 특정 화자에 맞는 마스크를 생성하는 연구가 제안되었다[30]. 화자 임베딩 네트워크의 경우 화자 검증(speaker verification)으로 학습한 네트워크를 사용했다. 화자 임베딩 네트워크에서 얻은 화자 임베딩을 음원 분리 네트워크의 장단기 메모리(Long Short-Term Memory, LSTM)층 입력 특징에 결합하여 컨디셔닝하는 방식을 제안한 것이 특징이다.

2.2.2 기타 분야 연구

본 절에서는 음악 음원 분리를 위해 제안되지는 않았지만 본 연구에서 제안한 네트워크의 기반이 되는 연구들과 해당 도메인에서 접근 방식이 본 연구와 유사한 연구들을 다룬다. 또한 다루고자 하는 문제가 유사한 연구들에 대해서 짚어본다.

본 연구에서 제안하는 주 입력을 처리하는 과정에서 보조로 주어지는 입력의 정보를 활용하는 네트워크의 구조와 가장 유사한 방식을 사용하는 분야로는 이미지 간 변환(image-to-image translation)를 꼽을 수 있다. 이미지 간 변환에 대표적인 태스크로는 스타일 전이(style transfer), 채색(colorization), 이미지 복원(inpainting) 등을 꼽을 수 있다. 이미지 간 변환 연구들 중 스타일 전이를 다루는 대부분의 연구들은 이미지를 구성하는 잠재 공간이 내용(content)와 스타일(style)로 나누어져있다는 가정을 한다. 이를 바탕으로 주어진 두 이미지 중 한 이미지에서 스타일 정보를 담고 있는 잠재 벡터를 추출하여 다른 이미지에 삽입하는 방식의 구성을 갖고 있다. *Xun* 은 스타일 정보를 담고 있는 잠재 벡터를 컨디셔닝하기 위해 적응적 인스턴스 정규화(Adaptive Instance Normalization, AdaIN)을 제안한다[3]. 이 방식은 인스턴스 정규화(instance normalization)[8] 이후 스케일링(scaling), 바이어싱(biasing) 파라미터를 스타일 잠재 벡터로부터 생성하여 이용하도록 동작하는데 컨디셔닝 효과가 뛰어남을 보였다.

적대적 생성 신경망(Generative Adversarial Network, GAN)[33]을 이용하는 이미지 간 변환 연구들 중 변환되는 이미지가 일대일(one-to-one) 대응이 아닌 일대다(one-to-many) 대응을 가질 수 있는 경우에서 보조 입력인 잠재 벡터의 변화에도 적은 수의 결과만 나타나는 모드 붕괴(mode collapse)가 나타나는 경우가 많은데 이를 해결하기 위해 구조적으로 접근한 연구도 있다[29]. Conditional VAE-GAN과 Conditional Latent Regressor-GAN을 결합한 구조로 학습하여 변환된 이미지를 다시 인코딩했을 때 보조 입력으로 주어진 잠재 벡터를 다시 복원하게끔 하는 방식을

제안했다. 이러한 방식은 잠재 벡터가 변환된 결과물에 반영되게하는 효과를 보여주었다.

한편 오디오 분야에서도 스타일 전이에 대한 연구가 활발하게 진행되고 있다. 또한, 스타일 전이 연구들에서 스타일은 음악의 장르, 악기의 음색 등 다양하게 정의되고 있다. 음악 장르 변환을 위해 CycleGAN[34]을 이용하여 MIDI 도메인에서 곡의 구조를 유지한 채 서로 다른 장르로 바뀌도록하는 방식을 제안했다[35]. 음악적 구조를 유지시키기 위해 추가적인 판별 모델(discriminator)를 두었으나 CycleGAN 특성상 정해진 두 장르 간의 변환만 이루어지는 것이 한계점이다.

이미지 간 변환 연구들과 비슷한 접근으로 목소리의 스타일을 전이하는 연구도 제안되었다[36]. 발화의 내용을 인코딩하는 네트워크와 스타일을 인코딩하는 네트워크를 따로 두고 두 정보를 이용하여 스펙트로그램으로 생성하는 디코더로 이루어진 구조를 제안했다. 학습 시에는 같은 사람의 두 발화를 이용하여 한 발화를 복원할 때 다른 발화의 스타일을 이용하는 방식을 제안했으며 발화의 정보를 인코딩할 때 bottleneck의 크기를 조절하는 것으로 내용과 스타일의 분리를 이끌어냈다. 이러한 방식은 쌍으로 이루어진 데이터가 없어도 학습이 가능하며 zero-shot 전이도 가능하다는 점에서 의의가 있다.

딥러닝에 사용되는 데이터셋들 중 데이터 수집이 어려운 경우 크기가 매우 한정적인 경우가 많으며 음원 분리에 사용되는 데이터셋들[10, 14]도 그 중 하나다. 이처럼 데이터가 적은 경우 학습한 네트워크는 데이터셋에 과적합되기가 쉬우며 학습한 데이터셋과 다른 환경의 데이터에 대해서는 잘 동작하기 어렵다. 이를 해결하기 위해 이미지 분야에서는 데이터를 늘리기 위해 이미지를 회전(rotation), 이동(shifting), 반전(mirroring), 확대 또는 축소(scaling)등의 방법을 보편적으로 사용한다. 또한 *Zhang*은 학습 시에 서로 다른 샘플 간의 선형 보간을 통해 만들어낸 데이터를 학습한 경우 모델의 일반화가 더 잘 이루어짐을 보였다[37]. 이와 같은 맥락으로

[25]에서 학습 과정에서 음원 분리를 수행할 혼합 음원에 들어갈 소스들을 각각 임의의 곡에서 임의의 부분을 가져와 구성하는 방식을 사용했다. 이 경우에도 혼합 음원은 일반적인 음악적 특성을 전혀 고려하지 않았음에도 효과가 있음을 보였다.

제 3 장 제안 기법

본 장에서는 제안된 오디오 쿼리 기반 음원 분리 프레임워크에 대한 설명을 다룬다. 제안 프레임워크를 이루는 딥러닝 모듈들의 구성과 역할에 대한 내용과 함께 네트워크의 학습 과정에서의 데이터 구성 방법과 사용한 목적 함수 그리고 테스트 과정에 대해 서술한다.

3.1 오디오 쿼리 기반 음원 분리

제안한 오디오 쿼리 기반 음원 분리 프레임워크는 오디오 쿼리 인코딩 네트워크(query-net) $\mathbb{Q}(\cdot)$와 음원 분리 네트워크(separator) $\mathbb{S}(\cdot)$로, 두 개의 딥러닝 모듈로 구성된다. 기존에 제안된 대부분의 음원 분리 연구들은 통상적으로 $\mathbb{S}$만을 사용하여 음악에서 한 가지의 정해진 악기의 신호만을 분리하도록 되어있다. 이에 더해서, 제안한 프레임 워크는 별도의 입력인 오디오 쿼리을 이용하여 음악에서 특정 악기의 신호를 분리하도록 $\mathbb{Q}$가 추가되었다. 제안 기법의 전체 개요는 그림 3.1에서 보여지

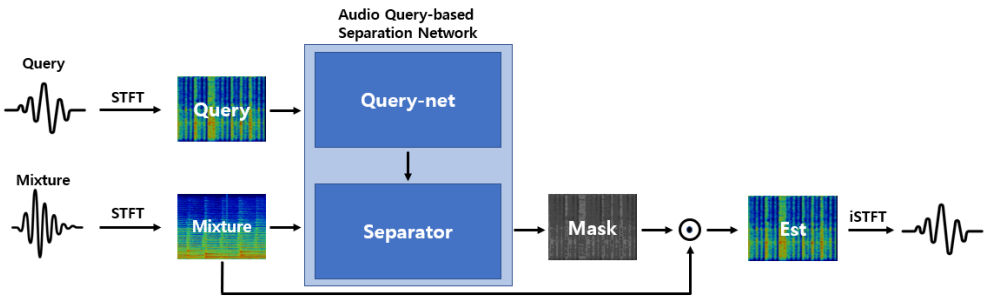


그림 3.1: 제안 프레임워크 개요도

는 것과 같다. 이러한 방식은 음악에서 분리할 악기를 오디오 쿼리을 통해 조절할 수 있도록 해주며, 따라서 오디오 쿼리로 주어지는 신호는 완전히 일치하지는 않아도 분리하려는 신호와 비슷해야 한다. $\mathbb{S}$의 분리 결과가 잠재 벡터(latent vector)에 따라 조절되게 하기 위해 우선 $\mathbb{Q}$가 주어진 오디오 쿼리을 잠재 공간의 벡터로 압축한다.

$\mathbb{Q}$는 6개의 합성곱(convolutional) 층과 1 층의 게이트 순환 유닛(gated recurrent unit)으로 구성되어 있다. 합성곱 층은 stride 합성곱으로 되어 입력이 각 층을 지날수록 입력의 크기는 작아지며, 주어진 입력에서 로컬 특징을 추출하는 역할을 한다. 합성곱 층을 지난 특징은 채널 축으로 쌓여진 것을 주파수 축으로 쌓아올려 형태가 바뀐다. 채널 축으로 쌓아 올려진 특징은 게이트 순환 유닛 층에 넘겨지며 마지막 상태의 값을 취한다. 또한 $\mathbb{Q}$가 오디오 쿼리의 정보를 잘 압축하고 유의미한 정보만을 담게하기 위해 인코딩된 벡터의 크기를 입력에 비해 작게 설정했다. 위 과정을 통해 오디오 쿼리의 정보가 압축된 벡터는 $\mathbb{S}$로 전달된다.

$\mathbb{S}$는 여러 음원 분리 연구[5, 13, 21, 23, 24]에서 성능이 입증된 U-net[16]을 기반으로 한다. U-net은 합성곱 층으로 이루어진 인코더와 역 합성곱(deconvolutional) 층으로 이루어진 디코더로 구성되며 각 층간 스킵 연결(skip-connection)을 가진 구조로 되어 있다. $\mathbb{S}$는 음원과 $\mathbb{Q}$가 오디오 쿼리을 인코딩한 잠재 벡터 $\mathbf{z}$를 입력으로 받아 음원 분리를 위한 소프트 마스크를 생성한다. 압축된 오디오 쿼리에 대한 정보를 마스크 생성하는 과정에 효율적으로 전달하기 위해 두 가지의 컨디셔닝 방법을 사용한다. 첫째로, $\mathbf{z}$를 입력으로 주어진 음원과 결합하여 U-net의 입력으로 준다. 이 과정에서 결합은 그림 3.2에서 볼 수 있듯이 $\mathbf{z}$를 음원의 채널 축으로 쌓아서 한다. 두번째로, $\mathbb{S}$의 디코더 각 층에 적응적 인스턴스 정규화(Adaptive instance normalization, AdaIN)[3]를 적용한다. 적응형 인스턴스 정규화는 잠재 벡터를 컨디셔닝하는데 유용한 것으로 여러 논문에서 보여진 바가 있다[3, 6]. 적응형 인스턴스 정규화는 역 합성곱을 거친 후 활성화 함수를 지나기 전의 텐서 $\mathbf{x}$에 두 연산이 적용되는 것으로 이루어진다. 첫 단계는 각 i번째 특징 $\mathbf{x}_i$에 인스턴스 정규화(instance

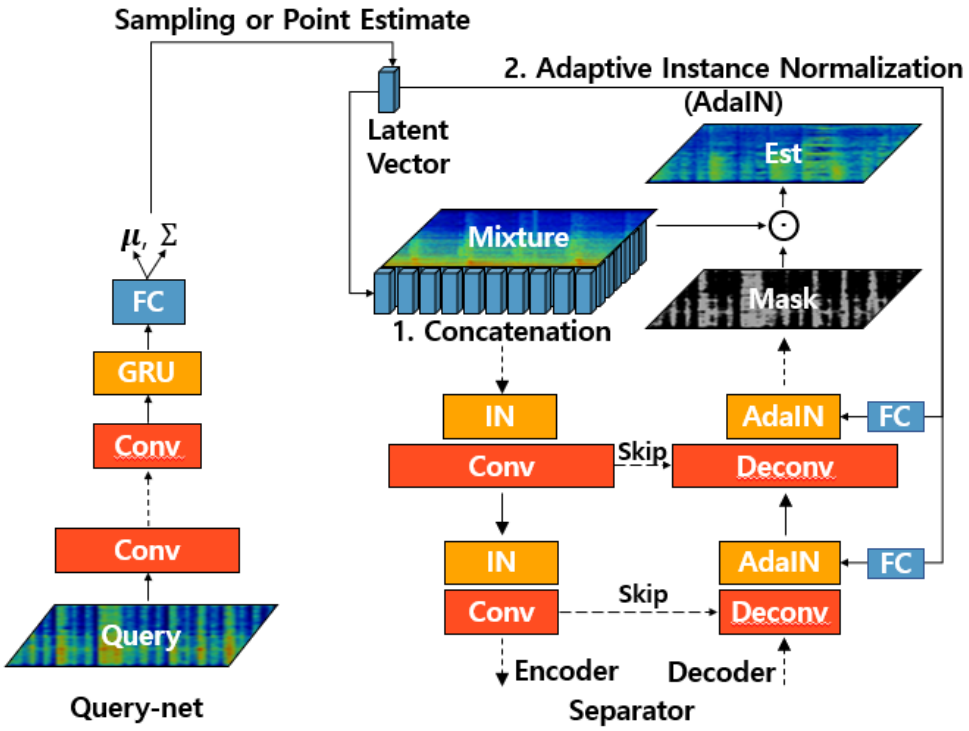


그림 3.2: 음원 분리 네트워크 컨디셔닝 방법

normalization)를 적용한다[8]. 다음으로, 학습된 파라미터를 이용하여 $\mathbb{Q}$에서 얻어진 $\mathbf{z}$를 $\mathbf{y}_s$와 $\mathbf{y}_b$로 변환한다.

$$\mathbf{y}_s = W_s^T \mathbf{z}, \quad \mathbf{y}_b = W_b^T \mathbf{z} \tag{3.1}$$

식 3.1에서의 W_s, W_b는 학습 가능한 파라미터를 의미한다. $\mathbf{y}_s$와 $\mathbf{y}_b$를 이용하여 앞서 인스턴스 정규화가 적용된 특징을 변환한다.

$$\text{AdaIN}(\mathbf{x}_i, \mathbf{y}) = \mathbf{y}_{s,i} \cdot (\frac{\mathbf{x}_i - \mu(\mathbf{x}_i)}{\sigma(\mathbf{x}_i)}) + \mathbf{y}_{b,i}. \tag{3.2}$$

3.2 학습

3.2.1 학습 데이터 구성

본 절에서는 학습 단계에서 분리하고자 하는 소스(source)와 해당 소스가 포함된 음원 m (mixture)을 구성하는 과정에 대한 설명을 다룬다.

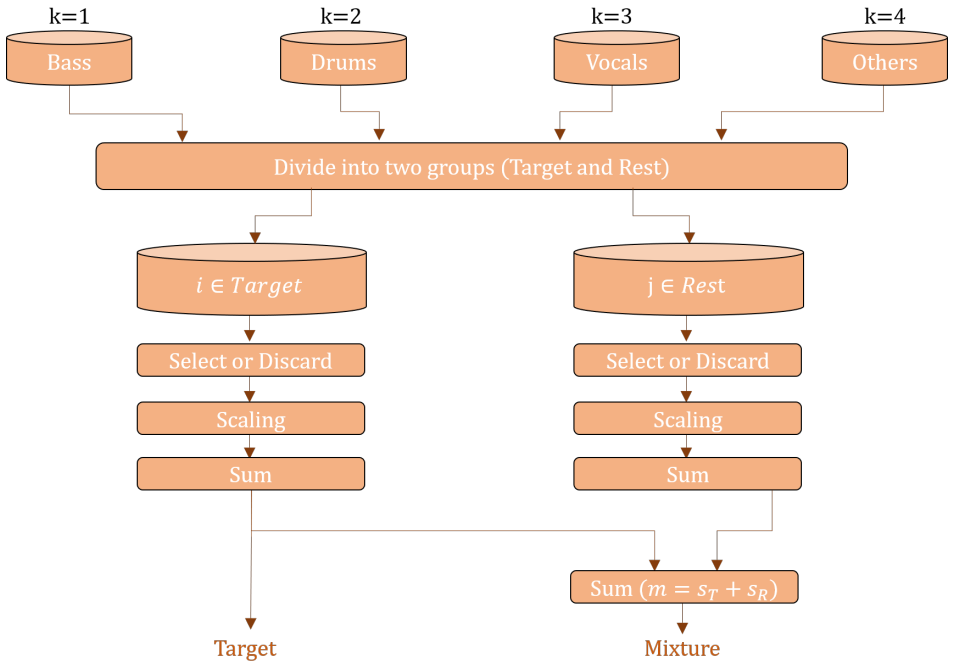


그림 3.3: 학습 데이터 구성 흐름도

우선 설명을 위해 데이터 셋에 총 K개의 소스 클래스가 있을 때, i 번째 단일 소스에서 샘플링된 신호를 v_i로 나타낸다. 이때 K가 5 이상이 되는 경우 구성되는 혼합 음원에 너무 많은 악기가 포함되는 것을 막기 위해 구성하는 단계마다 4개의 소스 클래스 만을 우선 샘플링한다. 이후 학습 데이터를 구성하기 위해, 각 악기를 무작위로 분리하고자 하는 그룹 T (target) 또는 나머지 그룹 R (rest) 중 하나에 포함되도록 나눈다. 이후 한 그룹에 너무 많은 악기가 포함되지 않게 하기 위해 베르누이 분포에서 샘플링 된 이진수 α_i를 ($\alpha_i \sim Bernoulli(0.5)$) v_i에 곱한다. 추가로 데이터 증가 기법의 일환으로 [25], 연속균등분포에서 샘플링한 값 β_i를 ($\beta_i \sim \mathcal{U}$[0.25, 1.25]) 소스 v_i에 곱해준다. 마지막으로 각 그룹별로 속한 소스들을 합쳐서 두 웨이브폼의 신호 s_T, s_R을 얻을 수 있다. 마지막으로 m은 식 3.3과 같이 s_T와 s_R의 선형 합으로 만들어진다.

$$m = s_T + s_R = \sum_{i \in T}(\beta_i \cdot \alpha_i \cdot v_i) + \sum_{j \in R}(\beta_j \cdot \alpha_j \cdot v_j). \tag{3.3}$$

제안한 네트워크의 각 모듈은 스펙트로그램을 입력으로 받기 때문에 m, s_T, s_R은 단시간 푸리에 변환(short-time-Fourier-transform, STFT) 도메인으로 변환하여 사용하며 앞에서 언급한 순서대로 대문자인 M, S_T, S_R로 표현한다.

3.2.2 학습 목적

제안한 프레임워크를 설계하기 위해 본 연구에서는 conditional variational autoencoder(cVAE)를 차용했다. 일반적인 인코더에서 잠재 벡터 $\mathbf{z}$은 결정적으로 잠재 공간으로 인코딩될 수 있는 반면, cVAE 프레임워크에서는 $\mathbf{z}$가 가우스 분포에서 샘플링되며, 여기서 분포(평균 및 분산)의 파라미터는 $\mathbb{Q}$에서 추정된다. $\mathbb{S}$는 주어진 M과 $\mathbf{z} \sim \mathbb{Q}(S_T)$을 이용하여 S_T를 재구성한다. 이는 cVAE의 두 목적 함수 중 하

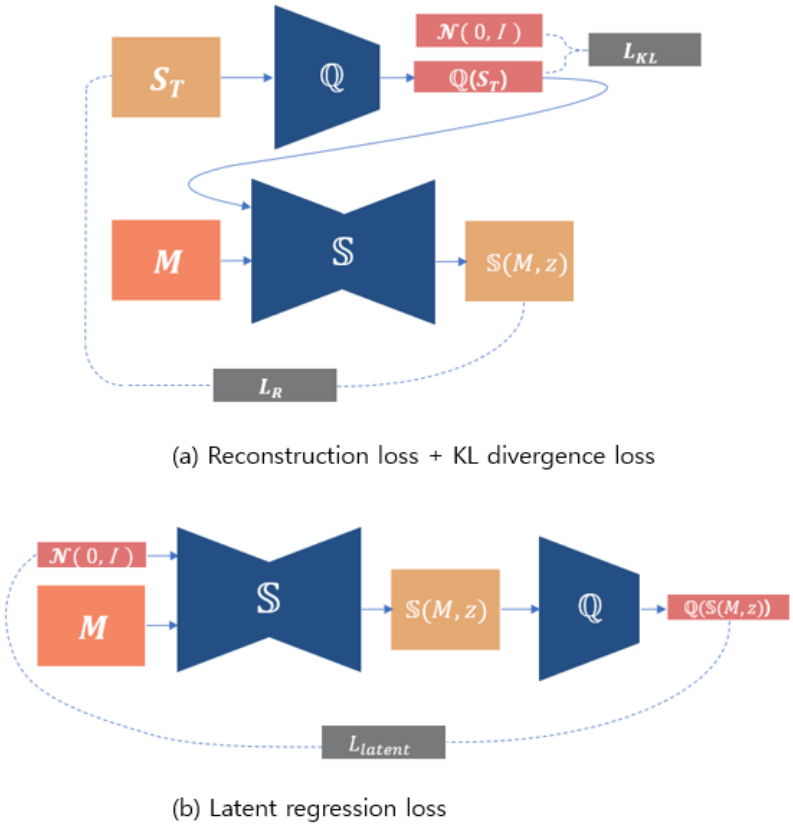


그림 3.4: 제안 네트워크 손실 함수

나인 reconstruction loss $\mathcal{L}_R$이며, 식 3.4와 같이 $\mathbb{S}$의 결과가 인코딩된 벡터에 의해 컨디셔닝되도록 한다.

$$\mathcal{L}_R = \mathbb{E}_{S_T \sim p(S_T),\, M \sim p(M),\, \mathbf{z} \sim \mathbb{Q}(S_T)}[\|S_T - \mathbb{S}(M, \mathbf{z})\|]_1 \tag{3.4}$$

학습 과정에서는 역전파(backpropagation)을 위해 잠재 벡터 $\mathbf{z}$를 샘플링하여 re-parameterization trick을 사용한다[7].

다음으로 $\mathbf{z}$의 분포가 가우시안 분포 $\mathcal{N}(\mathbf{0}, I)$와 가까워지도록 KL-divergence 손실 함수가 사용된다.

$$\mathcal{L}_{\mathrm{KL}} = \mathbb{E}_{S_T \sim p(S_T)} \left[\mathcal{D}_{\mathbf{KL}}(\mathbb{Q}(S_T) \| \mathcal{N}(\mathbf{0}, I)) \right] \tag{3.5}$$

cVAE 프레임워크와 더불어, $\mathbb{S}$의 출력이 잠재 벡터 $\mathbf{z}$에 의해 컨디셔닝되는 효과를 높이기 위해 [29]에서 사용된 latent regressor의 목적 함수도 적용했다. 우선 가우시안 분포 $\mathcal{N}(\mathbf{0}, I)$에서 샘플링한 임의의 벡터 z를 $\mathbb{S}$의 입력으로 전달한다. 따라서 $\mathbb{S}$는 z의 정보가 반영된 결과를 출력하며 이 결과로부터 $\mathbb{S}$를 이용하여 z를 복원하도록 하도록 한다. 여기서 식 3.4, 식 3.5과는 달리 $\mathbb{S}$의 출력 중 평균 값 (μ)만을 이용하여 $\mathbf{z}$를 복원한다.

$$\mathcal{L}_{latent} = \mathbb{E}_{M \sim p(M),\, \mathbf{z} \sim p(\mathbf{z})} \| \mathbf{z} - \mathbb{Q}(\mathbb{S}(M, \mathbf{z})) \|_1 \tag{3.6}$$

최종적으로, 전체 목적 함수는 다음과 같다.

$$\mathcal{L}_{Total} = \lambda_R \mathcal{L}_R + \lambda_{\mathrm{KL}} \mathcal{L}_{\mathrm{KL}} + \lambda_{latent} \mathcal{L}_{latent} \tag{3.7}$$

3.3 테스트

학습 단계에서는 $\mathbb{S}$가 식 3.4에서와 같이 분리하려는 소스를 오디오 쿼리으로 사용하여 분리하도록 학습되었다. 하지만 학습 과정과 달리 테스트 단계에서는 음원에서 분리하려는 소스는 따로 얻을 수 없기 때문에 오디오 쿼리과 분리하려는

음원이 다른 문제가 발생한다. 그럼에도 불구하고, $\mathbb{Q}$에서 인코딩되는 $\mathbf{z}$의 차원을 작게 학습하기 때문에 악기 클래스와 같은 고차원의 정보를 담도록 학습되었다고 할 수 있다. 따라서 테스트 단계에서 이러한 특성을 이용할 수 있다. 예를 들어, 사용자가 음원에서 특정 신호를 분리하고 싶을 때, 특정 신호와 완전히 같지는 않지만 비슷한 음악적 특성을 가진 소량의 샘플을 구하는 것은 가능하다. 따라서 소량의 샘플을 오디오 쿼리으로 이용하여 오디오 쿼리 인코딩 네트워크를 통해 잠재 벡터로 인코딩하고 음원 분리 네트워크는 그에 맞는 특정 신호를 추출할 수 있다.

또한, 오디오 쿼리 기반의 접근 방식과는 별도로, 학습 데이터셋에 포함된 각 소스 클래스의 데이터를 잠재 벡터로 인코딩한 뒤 소스별로 평균을 취하여 단일 클래스의 일반적인 특성을 반영하는 벡터를 얻을 수 있다. 따라서 위 과정으로 얻은 평균 벡터를 음원 분리 네트워크 $\mathbb{S}$로 직접 전달한다면 오디오 쿼리가 없어도 일반적인 음악 음원 분리 네트워크처럼 동작할 수 있음을 의미한다.

제 4 장 실험

본 장에서는 본 연구에서 제안한 오디오 쿼리 기반 음원 분리 프레임워크가 다양한 상황에서 유용하게 쓰일 수 있음을 보이는 실험과 결과를 다룬다. 실험에 사용한 학습 데이터 셋과 실험 설정의 세부 사항과 더불어 실험 내용으로는 오디오 쿼리을 이용한 특정 악기 분리, 잠재 벡터 보간(interpolation)을 이용한 음원 분리, 잠재 벡터가 음원 분리 성능에 미치는 영향에 대한 분석, 분리 반복법, 다른 알고리즘과의 성능 비교를 서술한다.

4.1 데이터셋

네트워크의 실험을 위한 데이터셋으로는 MUSDB18 데이터셋과 Slakh 데이터셋을 이용했다. MUSDB18 데이터셋은 트레이닝셋 100곡과 테스트셋 50곡으로 나뉘며 각 곡은 44.1kHz의 샘플링 율(sampling rate), 스테레오 포맷으로 녹음되었다. 데이터셋은 각 곡을 구성하는 소스를 '보컬(vocals)', '드럼(drums)', '베이스(bass)', '기타(other)'로 거칠게 정의된 클래스로 나누어 제공한다. 특히 '기타'의 경우 '보컬', '드럼', '베이스'를 제외한 모든 악기를 하나의 클래스로 묶여있다.

Slakh 데이터셋은 MUSDB18 데이터셋과 다르게 MIDI 파일을 렌더링한 곡으로 구성되어 있지만 악기의 종류와 곡의 수가 큰 것이 특징이다. 트레이닝셋 1500곡, 밸리데이션셋 375곡, 테스트셋 225곡, 총 2100곡으로 구성되어 있으며 각 곡은 44.1kHz의 샘플링 율, 모노 포맷으로 되어있다. 각 곡을 구성하는 소스를 제공하며 각 소스의 클래스를 표4.1과 같이 넓은 범위와 더 세밀한 범위로 나누어 제공해주는 점이 특징이다.

표 4.1: Slakh 데이터셋 악기 분류표

Coarsely Defined Labels							
Piano	Bass	Guitar	Strings	Synth Pad	Pipe	Brass	Drums
Finegrained Labels							
Electric Piano 1	Electric Bass(finger)	Distortion Guitar	Violin	Pad 1	Pan Flute	Brass Section	Drums
Electric Piano 2	Electric Bass(pick)	Electric Guitar(jazz)	Orchestral Harp	Pad 2	Flute	Trumpet	
Electric Grand Piano	Fretless Bass	Electric Guitar(muted)	Timpani	Pad 3	Whistle	French Horn	
Acoustic Grand Piano	Acoustic Bass	Electric Guitar(clean)	Cello	Pad 4	Piccolo	Trombone	
Bright Acoustic Piano	Slap Bass 1	Acoustic Guitar(steel)	Pizzicato Strings	Pad 5	Ocarina	Muted Trumpet	
Clavinet	Slap Bass 2	Acoustic Guitar(nylon)	Tremolo Strings	Pad 6	Recorder	Tuba	
Honky-tonk Piano		Guitar harmonics	Contrabass	Pad 7	Shakuhachi		
Harpsichord		Overdriven Guitar	Viola	Pad 8	Blown Bottle		

본 장에서 진행하는 실험 중 4.7절은 Slakh 데이터셋으로 학습한 네트워크를 사용했으며 이를 제외한 실험은 모두 MUSDB18 데이터셋을 이용해 학습한 네트워크로 진행되었다.

학습을 위해 두 데이터 셋에서 모두 각 곡은 22050Hz로 재샘플링(resampling)되었으며, 네트워크의 입력 크기인 3초 단위로 분할했다. 분할된 오디오에 단시간 푸리에 변환을 1024의 크기를 가진 윈도우가 75% 중복되도록 적용하여 스펙트로그램을 얻었다. 분리된 결과를 오디오로 복원하기 위해 합성 음원의 위상을 이용한 역단시간 푸리에 변환(inverse short-time-Fourier-transform)을 적용했다. 본 연구에서 제안한 네트워크는 단일 채널의 오디오를 입력으로 받기 때문에 MUSDB18에서는 테스트 결과를 얻기 위해 각 음원의 좌우 채널을 나누어 분리한 뒤 다시 스테레오로 재구성하는 방식을 이용했다. 또한 22050Hz의 결과물을 44.1kHz로 재샘플링했다. Slakh 데이터셋을 이용한 실험의 경우 모노로 제공되기 때문에 분리 결과물을 44.1kHz로 재샘플링하는 과정만 밟았다. 정량적 평가를 위한 척도인 신호 대 왜곡 비율(signal to distortion ratio, SDR)을 얻기 위해 museval 패키지[1]를 사용했다. SDR은 다음과 같이 계산된다.

[1] https://sigsep.github.io/sigsep-mus-eval

$$SDR = 20\log_{10}\left(\frac{\|s_{\text{target}}\|}{\|s_{\text{interf}} + s_{\text{artif}}\|}\right) \tag{4.1}$$

식 4.1에서 s_{target}는 목표 신호, s_{interf}는 다른 음원의 간섭 정도, s_{artif}는 복원 과정에서 생기는 노이즈를 의미한다. 이 외에도 신호 대 간섭 비율(signal to interferences ratio, SIR), 신호 대 결함 비율(signal to artifacts ratio, SAR)이 있으나 SDR이 종합적인 성능을 나타내는 지표기 때문에 본 연구에서는 SDR만을 이용한다.

4.2 실험 상세 설정

본 절에서는 실험에 사용된 네트워크의 상세 정보와 학습과 관련된 파라미터에 대한 설명을 다룬다. 각 네트워크의 구조는 그림 4.1와 같다. 그림 4.1에서 Conv는 합성곱 층을 의미하며 괄호 안의 숫자는 (필터의 주파수 축 크기, 필터의 시간 축 크기, 필터의 주파수 축 stride, 필터의 시간 축 stride, 필터의 수)의 순서로 표시되었다.

$\mathbb{Q}$는 4×4 크기의 필터를 가진 6 개의 합성곱 층과 게이트 순환 유닛으로 구성되어 있으며 각 층의 출력은 순서대로 32, 32, 64, 64, 128. 128의 아웃풋 채널을 가진다. 매 합성곱 층 이후에 인스턴스 정규화(instance normalization)와 ReLU 활성화 함수가 적용되었다. 게이트 순환 유닛의 유닛은 128개로 사용했다. 마지막으로 분포의 모수를 출력하기 위한 전결합층(fully connected layer)가 자리한다.

$\mathbb{S}$는 인코더와 디코더로 된 U-net 구조를 기반으로 하고있으며 인코더의 첫 번째 층과 디코더의 마지막 층을 제외하고 모두 시간 축으로 stride 크기 2를 가지며 각 인코더 층은 합성곱 인스턴스 정규화와 leaky ReLU가 적용되었다. 디코더 층은 적응적 인스턴스 정규화와 leaky ReLU가 적용 되었다. 마지막 층의 경우 소프트 마스크를 생성하기 위해 leaky ReLU 대신 sigmoid가 사용되었다.

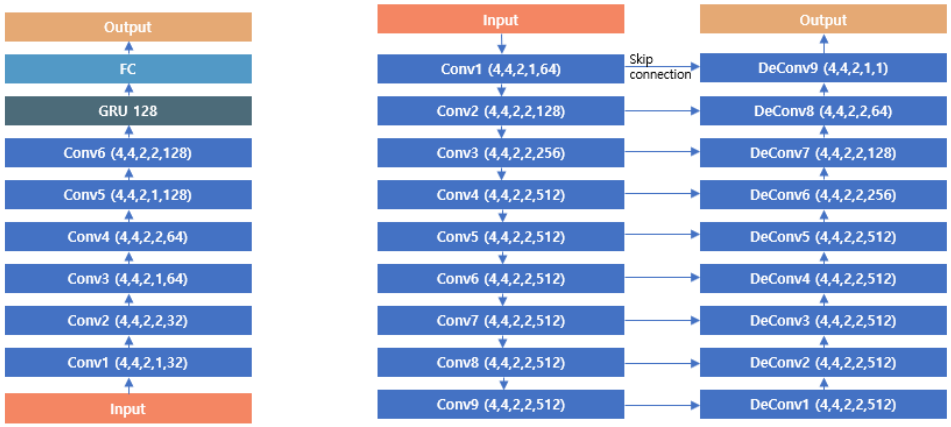


그림 4.1: 네트워크 상세 구조

잠재 벡터의 차원은 32로 설정했으며, 배치(batch) 크기는 5로 학습했다. 식 3.7에서의 계수로 $\lambda_R = 10$, $\lambda_{\mathrm{KL}} = 0.01$, $\lambda_{latent} = 0.5$를 사용했다. 초기 학습률(learning rate)은 0.0002로 설정했으며 200000 이터레이션(iteration) 이후 매 10000 이터레이션마다 5×10^{-6} 씩 감소했다. 최적화 함수(optimize function)로는 $\beta_1 = 0.5$, $\beta_2 = 0.999$ 로 설정한 Adam[19]을 사용했다.

4.3 새로운 샘플에 대한 쿼리 인코딩 네트워크 동작

오디오 쿼리 기반 음원 분리를 수행하기 위해선 쿼리 인코딩 네트워크가 학습 중에는 보지 못했던 샘플들에 대해서도 강인하게 동작해야 한다. 새로운 샘플들의 음악적 특성을 잘 잡아내는지 확인하기 위한 방법으로 본 실험에서는 테스트셋에 포함된 소스들을 이용한다. 'Vocals','Drums','Bass' 그리고 'Other'로 나뉘어진 소스들을 네트워크 입력 단위로 자른 뒤 스펙트로그램으로 변환하여 쿼리 인코딩 네

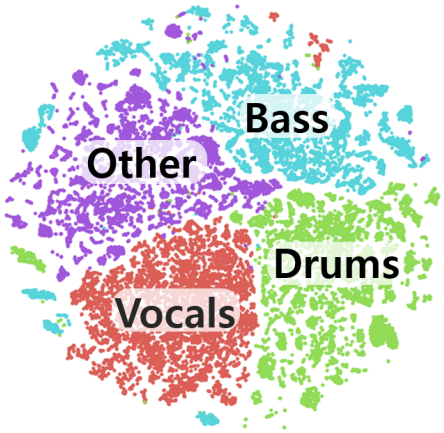


그림 4.2: 테스트셋 샘플의 잠재 벡터 t-SNE 시각화

트워크를 이용해 잠재 벡터로 변환한다. 변환된 잠재 벡터들을 관찰하기 위해 t-SNE(t-Stochastic Nearest Neighbor)를 이용해 시각화했다[11]. t-SNE는 고 차원의 데이터를 저 차원의 데이터로 고 차원에서의 거리 관계를 유지하며 차원을 축소시키는 알고리즘으로 고 차원의 데이터를 시각화하는데 널리 쓰인다.

시각화 결과는 그림 4.2에 나타나 있다. 학습 과정에서 별도의 분류 목적 함수를 사용하지 않았음에도 불구하고 쿼리 인코딩 네트워크는 처음 보는 샘플들에 대해 서로 다른 점으로 맵핑하면서도 각 소스 별로 유사한 공간으로 인코딩한 것을 볼 수 있다.

4.4 오디오 쿼리를 이용한 특정 악기 분리

본 연구에서 제안한 프레임워크가 주어진 오디오 쿼리의 특성을 포착하여 그에 따라 분리하는 것을 검증하기 위해 특정 악기를 분리하는 실험을 실시했다. 그림 4.3

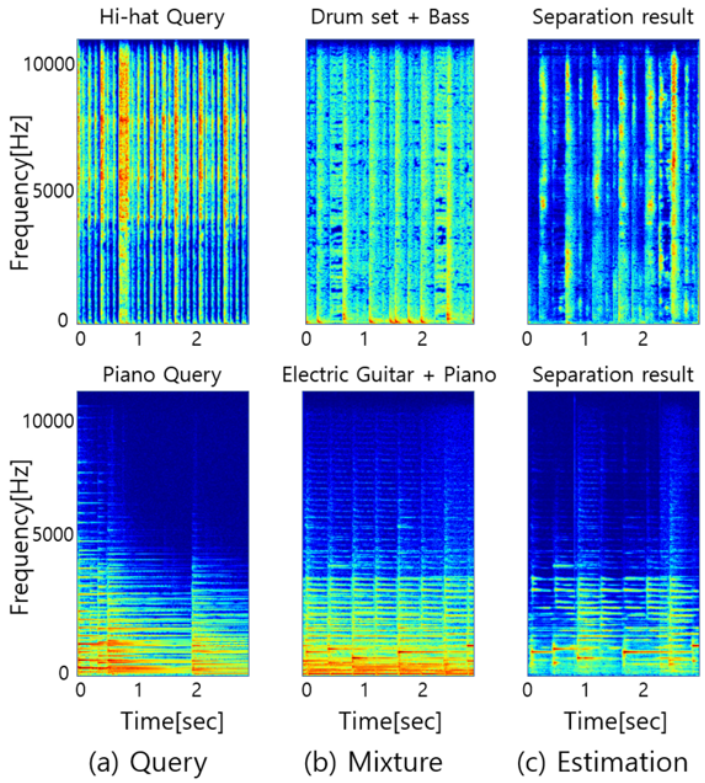


그림 4.3: 오디오 쿼리를 이용한 특정 악기 분리

에서와 같이, (하이햇 + 킥 드럼 + 베이스)와 (피아노 + 일렉트로닉 기타)의 합성 음원에 하이햇과 피아노의 오디오 쿼리가 주어졌다. 쿼리와 합성 음원은 트레이닝 셋에 포함되지 않은 곡을 사용했으며, 두 쿼리는 합성 음원이 아닌 다른 곡에서 얻었다. 하이햇 분리 결과에서 하이햇을 이루는 주파수 대역의 성분이 남아 있는 반면, 킥 드럼과 저주파 대역에 있는 베이스가 대부분 제거된 것을 관찰할 수 있다. 피아노 분리의 결과는 하이햇의 경우처럼 명확하지는 않지만 일렉트로닉 기타가 상당히 제거된 것을 확인할 수 있다.

이 실험에서 주목할 사실은 네트워크가 '보컬', '드럼', '베이스', '기타' 이상의 세밀한 계층적인 악기 클래스 정보가 없는 MUSDB18 데이터셋으로 학습했다는 점이다. 데이터셋이 제공하는 클래스의 정의에 따라 하이햇과 킥드럼은 '드럼'으로, 피아노와 일렉트로닉 기타는 '기타' 클래스로 분류된다. 그럼에도 불구하고, 본 연구에서 제안한 네트워크는 하이햇과 피아노를 분리해내는 결과를 보여주었고, 이는 제로샷(zero-shot) 음원 분리라고 할 수 있다. 이러한 결과는 제안된 방법이 오디오 쿼리 기반 음원 분리에 잘 적용될 수 있음을 나타낸다.

4.5 잠재 벡터 보간을 이용한 음원 분리

본 절에서는 각 악기의 평균 벡터를 구한 뒤 평균 벡터 간의 보간을 하여 음원 분리에 사용하는 실험을 진행했다. 평균 벡터를 구하기 위해 식 4.2과 같이 트레이닝셋에 포함된 곡들을 3초 단위로 나누어 벡터들을 구한 뒤 평균을 계산했다.

$$\mathbf{z}_c = \frac{1}{N_c} \sum_i \mathbb{Q}(S_{c,i}) \tag{4.2}$$

식 4.2에서 $S_{c,i}$는 악기 클래스 c에 포함된 i 번째 3초 길이의 스펙트로그램을 뜻하며, N_c는 악기 클래스 c에 속한 스펙트로그램의 총 수를 뜻한다.

보간법으로는 [28]에서 제안된 구면 선형 보간법(spherical linear interpolation, ($Slerp$)를 사용했다.

$$Slerp(\mathbf{z}_1, \mathbf{z}_2; \alpha) = \frac{\sin(1-\alpha)\theta}{\sin\theta}\mathbf{z}_1 + \frac{\sin\alpha\theta}{\sin\theta}\mathbf{z}_2 \tag{4.3}$$

식 4.3에서 α는 보간 가중치를 나타내며 θ는 보간을 하는 두 벡터 $\mathbf{z}_1$와 $\mathbf{z}_2$ 사이의

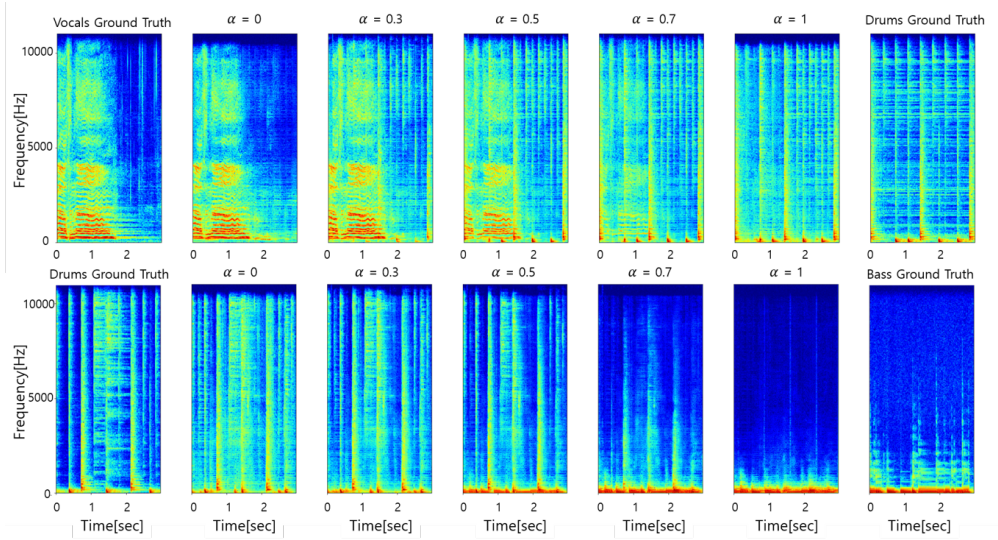


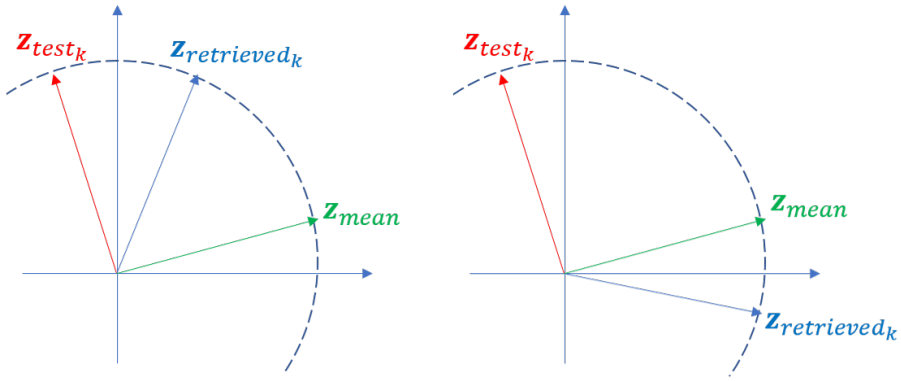
그림 4.4: 잠재 벡터 보간을 이용한 음원 분리

각도를 의미한다.

그림 4.4은 드럼 ($\mathbf{z}_{drums}$) → 베이스 ($\mathbf{z}_{bass}$), 보컬 ($\mathbf{z}_{vocals}$) → 드럼 ($\mathbf{z}_{drums}$) 두 악기의 평균 벡터를 보간하여 음원 분리한 결과를 보여주고 있다. 가중치 α에 따라서 분리 결과에 포함된 악기의 비율이 바뀌는 것을 스펙트로그램 상에서 확인할 수 있다. 따라서 본 실험의 결과는 제안한 방법이 잠재 공간을 조절함에 따라 연속적으로 바뀌는 분리 결과를 낼 수 있음을 보여준다.

4.6 잠재 벡터가 음원 분리 성능에 미치는 영향 분석

본 절에서는 분리에 사용되는 잠재 벡터에 따른 분리 성능의 변화를 관찰하고 어떤 경우에서 분리 성능이 높아지는 지에 대한 대한 실험을 다룬다. 실험을 위해 우선 데이터셋에 포함된 모든 보컬 트랙의 트랙별 평균 벡터를 식 4.4의 방법으로



(a) $\Delta CD > 0$ (b) $\Delta CD < 0$

그림 4.5: ΔCD의 경우

구한다.

$$\mathbf{z}_i = \frac{1}{N_i} \sum_j \mathbb{Q}(S_{i,j}) \tag{4.4}$$

트랙별 평균 벡터를 구하는 방식은 식 4.2과 비슷한 과정으로 각 곡을 3초 단위로 나뉘어 스펙트로그램으로 변환한다. 식 4.4의 $S_{i,j}$는 i 번째 보컬 곡의 j번째 스펙트로그램을 의미한다. N_i는 i번째 트랙에서 얻어진 3초 길이의 스펙트로그램 총 수를 뜻한다. 이후 트레이닝셋에 포함된 곡의 평균 벡터만을 이용해 트레이닝셋의 평균 벡터 $\mathbf{z}_{mean}$를 구한다.

$$\mathbf{z}_{mean} = \frac{1}{100} \sum_{i \in training} \mathbf{z}_i \tag{4.5}$$

마지막으로, 테스트셋에 포함된 곡들에 대하여 k 번째 테스트 보컬 트랙의 평균 벡터 $\mathbf{z}_{test_k} = \mathbf{z}_k,\ k \in test$와 트레이닝셋에서 코사인 거리($CD$)가 가장 가까운 곡의 평균 벡터 $\mathbf{z}_{ret_k}$를 구한다.

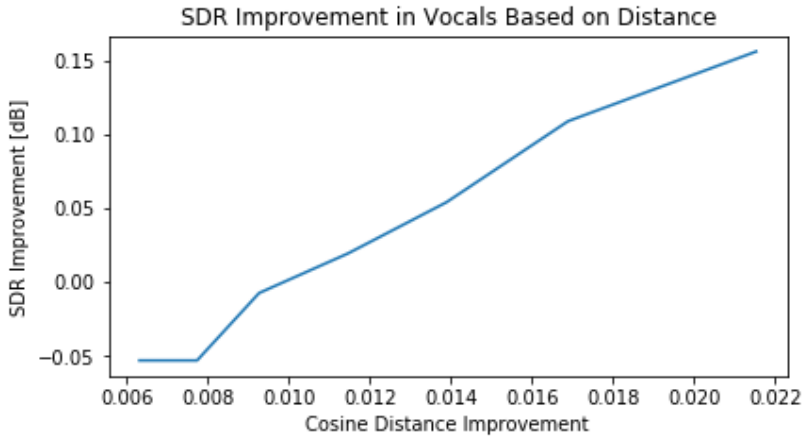


그림 4.6: ΔSDR과 ΔCD 간의 관계 그래프

$$CD(\mathbf{z}_1, \mathbf{z}_2) = 1 - \frac{\mathbf{z}_1 \cdot \mathbf{z}_2}{\|\mathbf{z}_1\| \, \|\mathbf{z}_2\|} \tag{4.6}$$

$$\tilde{k} = \underset{i \in training}{\arg\min} \; CD(\mathbf{z}_i, \mathbf{z}_{test_k}), \;\; \mathbf{z}_{ret_k} = \mathbf{z}_{\tilde{k}} \tag{4.7}$$

벡터에 따른 분리 성능 증가를 관찰하기 위해 두 경우로 나누어 테스트셋에서 보컬을 분리한다. 모든 테스트 곡에 대해 $\mathbf{z}_{mean}$을 이용해 얻은 결과 $\hat{S}_{mean} = \mathbb{S}(M, \mathbf{z}_{mean})$와 k 번째 테스트 곡에 대해 식 4.7를 통해 얻은 $\mathbf{z}_{ret_k}$를 이용해 $\hat{S}_{ret_k} = \mathbb{S}(M, \mathbf{z}_{ret_k})$를 얻은 결과를 비교한다. 신호 대 왜곡 비율(SDR)을 척도로 성능 증가를 다음과 같이 정의했다.

$$\Delta\text{SDR} = \text{SDR}(S_{GT_k}, \hat{S}_{ret_k}) - \text{SDR}(S_{GT_k}, \hat{S}_{mean}) \tag{4.8}$$

식 4.8에서 S_{GT_k}는 k 번째 정답(ground truth) 보컬 트랙의 스펙트로그램을 나타낸다. 분리에 사용된 잠재 벡터 간의 차이를 확인하기 식 4.6을 이용하여 $\mathbf{z}_{test_k}$

를 기준으로 코사인 거리를 측정한다. $(\mathbf{z}_{test_k}, \mathbf{z}_{ret_k})$ 사이의 거리와 $(\mathbf{z}_{test_k}, \mathbf{z}_{mean})$ 사이의 거리의 차이를 다음과 같이 구한다. 따라서 잠재 공간에서 $\mathbf{z}_{test_k}$를 음원 분리를 위한 가장 이상적인 잠재 벡터로 가정했을 때 잠재 공간 상에서 얼마나 거리가 개선되었는지를 측정하는 과정으로 설명할 수 있다.

$$\Delta CD = CD(\mathbf{z}_{test_k}, \mathbf{z}_{mean}) - CD(\mathbf{z}_{test_k}, \mathbf{z}_{ret_k}) \tag{4.9}$$

그림 4.5는 $\mathbf{z}_{ret_k}$를 사용할 때 발생 가능한 두 상황을 나타낸다. (a)는 ΔCD가 양인 경우로써 분리 성능 증가(ΔSDR > 0)가 있을 것이라 추정한다. 이 경우 $\mathbf{z}_{ret_k}$가 $\mathbf{z}_{mean}$보다 $\mathbf{z}_{test_k}$와 유사한 정보를 담고 있을 것으로 예상하여 나은 분리 결과를 기대할 수 있다. (b)는 음의 ΔCD인 경우로써 분리 성능이 낮아질 것으로 추정한다. (a)의 상황과 반대로, 분리 네트워크가 $\mathbf{z}_{test_k}$와 관련된 정보를 충분히 받지 못하기 때문에 분리 성능이 감소할 것으로 예상한다. 위의 가정을 실험적으로 보이기 위해, ΔSDR과 ΔCD 간의 관계를 그림 4.6의 그래프로 나타냈다. 그래프에서 분리에 사용된 잠재 벡터가 정답 벡터에 가까울수록 성능 증가 폭이 커지는 것을 확인할 수 있다. 이 실험 결과는 정답 벡터에 가까운 잠재 벡터를 사용할수록 더 높은 분리 성능을 기대할 수 있다는 가정을 뒷받침해준다.

4.7 세분화된 클래스 정보를 이용한 음원 분리 비교 실험

음원 분리를 하는 경우 분리하고자 하는 음원에 대해 보다 세밀한 정보가 주어진다면 더 높은 성능을 기대할 수 있을 것이라는 가정을 확인하기 위해 Slakh 데이터셋을 이용한 실험을 진행한다. Slakh 데이터셋에는 각 곡에 포함된 소스의 넓은 범주에서의 클래스 정보와 그보다 더 세밀한 범주에서의 클래스 정보를 함께 제공하기 때문에 넓게 정의된 클래스 정보를 활용했을 때와 더 세분화된 클래스 정보를

활용했을 때의 성능을 비교하기 용이하다.

실험을 진행하기 위해 4.4절의 실험과 유사한 방식으로 클래스별 평균 벡터를 구한다.

$$\mathbf{z}_c = \frac{1}{N_c} \sum_i \mathbb{Q}(S_{c,i}) \tag{4.10}$$

$$\mathbf{z}_f = \frac{1}{N_f} \sum_i \mathbb{Q}(S_{f,i}) \tag{4.11}$$

평균 벡터를 구하기 위해 각 곡을 네트워크의 입력 단위인 3초로 나누어 스펙트로그램으로 변환한다. 식 4.10, 식 4.11에서 c와 f는 순서대로 넓게 정의된 클래스 리스트와 세분화된 클래스 리스트에 포함된 클래스를 뜻한다. 또한 N_c, N_f는 해당 클래스에 포함된 스펙트로그램 S_c, S_f의 총 수를 나타낸다.

다르게 정의된 두 클래스 리스트에 포함된 클래스의 평균 벡터를 구한 뒤 k번째 테스트 곡에 대해서 넓은 범주의 클래스 평균 벡터를 사용한 결과 $\hat{S}_c = \mathbb{S}(M, \mathbf{z}_c)$와 좁은 범주의 클래스 평균 벡터를 사용한 결과 $\hat{S}_f = \mathbb{S}(M, \mathbf{z}_f)$를 비교한다. 앞선 절의 실험과 마찬가지로 신호 대 왜곡 비율의 차이를 구한다.

$$\Delta\text{SDR} = \text{SDR}(S_{GT_{k,c}}, \hat{S}_f) - \text{SDR}(S_{GT_{k,c}}, \hat{S}_c) \tag{4.12}$$

식 4.12에서 $S_{GT_{k,c}}$는 k번째 트랙의 c 클래스에 해당하는 소스의 스펙트로그램을 의미한다.

표 4.2에 각 클래스별로 성능 차이 중앙값이 기재되어 있다. 드럼의 경우 'Drums' 이상의 세분화된 클래스 정보가 제공되지 않기 때문에 본 실험에서는 제외되었다. 우선 모든 클래스에서 세분화된 평균 벡터를 사용했을 때 증가하거나 거의 비슷한

표 4.2: 클래스 정보 음원 분리 성능 비교

Source	Median ΔSDR
Bass	0.06
Piano	0.04
Guitar	0.45
Strings	0.72
Synth Pad	0.11
Pipe	0.01
Brass	0.61

것으로 관찰되었다. 가장 많이 오른 클래스로는 'Strings'와 'Brass'로 0.5dB 이상의 스코어가 증가했다. 반면 세분화된 클래스 정보로도 큰 차이가 없는 악기들도 보이는데 'Bass'와 'Synth Pad'의 경우 세분화된 악기 간에도 음색의 변화가 크지 않기 때문에 결과에서 큰 차이가 없는 것으로 추정된다. 한편 'Pipe'의 경우 세분화된 악기 간의 음색의 편차는 있지만 세분화된 악기들 중 'Flute'인 곡이 'Pipe' 클래스의 대부분을 차지하기 때문에 한 악기로 편중되어 나타난 결과로 해석된다. 본 절에서 진행한 실험은 악기마다 편차는 있지만 앞서 가정한 세분화된 악기의 정보를 활용하는 경우 더 높은 성능을 기대할 수 있다라는 가정을 뒷받침하는 결과를 보여준다.

4.8 분리 반복법

본 절에서는 음원 분리에 제안한 시스템을 반복적으로 적용하여 성능 증가를 꾀하는 방법에 대해 다룬다. 이후로 이러한 방법을 분리 반복법이라 칭하며, 분리 반복법은 다음과 같은 순서로 이루어진다. 우선, 특정 악기를 분리하기 위해 해당

표 4.3: 분리 반복법 성능 비교

	Vocals	Drums	Bass	Other
Single step	4.84	4.31	**3.11**	2.97
Iterative method	**4.90**	**4.34**	3.09	**3.16**

악기의 평균 벡터 $\mathbf{z}_{mean}$을 이용한다. 그 다음 분리된 결과물을 다시 $\mathbb{Q}$를 이용하여 인코딩한다. 재 인코딩된 벡터는 평균 벡터에 비해 정답 벡터에 더 가까울 것으로 기대할 수 있다. 따라서 분리 반복법의 마지막 단계로 재인코딩된 벡터를 분리에 사용하여 결과를 얻는다. 제안한 분리 반복법의 효과를 검증하고 분리하고자 하는 신호가 해당 클래스의 일반적인 특성과 다른 까다로운 상황에서 도움이 될 수 있음을 실험을 통해 보인다. 표 4.3은 MUSDB18 테스트셋에 대해 분리 반복법(iterative method)을 적용하지 않은 경우(single step)와 적용한 경우의 성능을 보여준다. 분리 반복법이 '드럼'과 '베이스'에서는 성능이 크게 바뀌지 않았지만 '보컬', '기타' 클래스에 대해서는 성능 증가가 있음을 확인할 수 있다.

결과 분석을 위해 분리 반복법을 통해 보컬 분리의 성능이 크게 증가한 곡들을 살펴보았다. 그 중 'Timboz - Pony'와 'Hollow Ground - Ill Fate'의 곡은 분리 반복법을 통해 SDR이 0.5dB 이상 증가했다. 두 곡의 보컬은 헤비 메탈 장르의 그로울링(growling)과 같은 기교를 사용하여 보컬의 일반적인 특성과는 거리가 멀다고 할 수 있다.

분리 반복법이 분리 목표 신호가 일반적인 특성과 다를 때 도움이 될 수 있다는 가정을 검증하기 위해 테스트셋에 포함된 소스들을 3초 단위로 분할한 뒤 잠재 벡터로 인코딩했다. 이후 잠재 벡터들을 분리 반복법을 통해 SDR이 0.4dB 이상 증가한 곡들에 포함된 벡터들과 그렇지 못한 그룹으로 나누었다. 두 그룹으로 나뉜 벡터들을 그림 4.7과 같이 t-SNE를 이용해 시각화한다. 그림 4.7에서 0.4dB 이상의 성능이

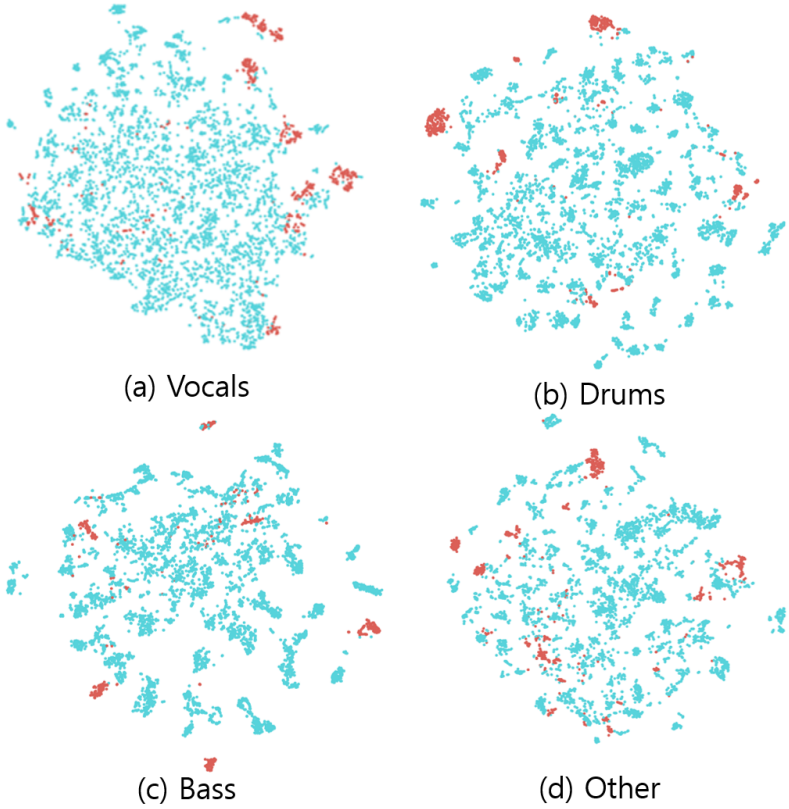


그림 4.7: 분리 반복법에 따른 성능 증가 곡 비교 t-SNE

증가한 그룹은 빨간점으로 표시되었다. 몇몇 벡터들은 중앙에 위치하지만 대부분은 외곽에 위치하고 있다. 이 벡터들은 이상치(outlier)들로 해석될 수 있으며 위 실험 결과는 특이한 성질을 가진 신호를 분리할 때 분리 반복법이 효과적임을 보여준다.

4.9 정량 평가

본 절에서는 제안한 기법의 성능을 정량적으로 평가하고 다른 음원 분리 알고리즘과 비교하기 위해 MUSDB18 데이터셋의 테스트셋으로 측정한 결과를 다룬다. 3장에서 서술한 바와 같이, 본 연구에서 제안한 프레임 워크는 오디오 쿼리를 직접 이용하여 음원을 분리하거나 앞서 진행한 실험처럼 여러 오디오 쿼리의 인코딩된 잠재 벡터들을 이용하여 평균을 내서 직접 음원 분리 네트워크에 입력으로 주는 방법으로도 음원 분리가 가능하다. 따라서 정량적 평가 또한 단일 오디오 쿼리를 사용한 경우와 분리에 쿼리가 필요하지 않은 알고리즘과 공정한 비교를 위해 평균 벡터를 분리에 이용한 경우의 결과로 나누어 진행한다. 또한 두 실험에서 모두 음원 분리는 단일 네트워크로 수행했다.

우선 오디오 쿼리를 이용한 음원 분리의 경우 테스트 시에 사용할 수 있는 오디오의 쿼리는 사용자의 선택에 따라 달라지며 앞선 실험에서 보였듯이 사용하는 오디오 쿼리에 따라 분리 결과의 성능이 달라지기 때문에 두 가지의 경우로 나누어 정량적 평가를 진행한다. 먼저 사용자가 분리하려는 신호와 유사한 오디오 쿼리를 구하기 어려워 분리하려는 신호와 제시한 오디오 쿼리의 유사성이 떨어지는 경우를 상정하기 위해 오디오 쿼리를 임의로 설정한 경우(Random Query, RQ)가 있다. 이 경우를 풀어서 설명하자면 k번째 ($k \in test$) 테스트 곡에서 c 클래스의 신호를 분리할 때 j번째($j \in test, j \neq k$) 곡에서 임의의 부분 $S_{c_{j,i}}$을 오디오 쿼리로 사용한다. $S_{c_{j,i}}$는 j번째 테스트 곡의 c 트랙을 네트워크 입력 단위인 3초 길이로 나누어 N개의 스펙트로그램으로 변환한 경우에 임의의 i번째 ($i \in \{1, 2, ..., N\}$) 스펙트로그램을 의미한다.

오디오 쿼리를 이용한 음원 분리의 두 번째 경우는 분리하려는 신호를 오디오 쿼리로 사용한 경우(Groundtruth Query, GQ)로 설정한다. 따라서 k번째 테스트 곡에서 c 클래스의 i번째에 위치한 신호를 분리할 때 오디오 쿼리로 $S_{c_{k,i}}$를 이용한 결과를

나타낸다. 따라서 오디오 쿼리를 이용한 음원 분리를 진행할 때 RQ를 성능의 하한선으로, GQ는 상한선으로 간주할 수 있다.

두 경우를 비교하기 위해 표 4.4를 살펴보면 당연하지만 정답지를 오디오 쿼리로 사용한 경우가 모든 클래스에서 높은 점수를 보이고 있다. 적게는 약 0.5dB에서 많게는 1dB 이상 차이가 나는 것을 확인할 수 있다. 이 비교 실험에서도 곡에 따라 크게 편차가 나지 않는 'Drums'와 'Bass'의 경우 차이가 약 0.5dB 가량으로 다른 클래스에 비해 적은 편을 기록했다. 반면 'Vocals'와 'Other'의 경우 곡별로 편차가 큰 편이기 때문에 그 두 경우에서 성능의 차이가 크게 나타났다. 특히 'Other'의 경우 가장 큰 차이를 보였는데 이는 여러 악기를 하나의 클래스로 정의하여 임의로 오디오 쿼리를 결정한 경우 분리하려는 신호와 다른 악기의 신호를 쿼리로 사용한 경우가 많이 발생할 수 있기 때문에 당연한 결과로 보인다.

다음은 평균 벡터(mean)를 이용한 음원 분리 실험에 대해 다룬다. 평균 벡터는 식 4.2와 동일한 방식으로 MUSDB18 트레이닝셋에 포함된 곡들을 인코딩하여 총 4개의 악기 클래스별 평균을 내서 음원 분리에 사용한다.

표 4.4에는 Ours로 표기된 제안 네트워크의 점수를 포함하여 SiSEC2018 [22]에 제출된 알고리즘들의 SDR의 중앙값(median) 점수가 표기되었다. 제안된 알고리즘이 가장 높은 성능을 기록하지는 못했지만, 성능 비교 결과는 단일 악기 또는 정해진 4개의 소스만 분리 가능한 다른 심층 학습 기반 알고리즘에 견줄 수 있음을 보여준다. 이는 제안한 방법이 오디오 쿼리 기반 음원 분리에 한정되는 것이 아니라 오디오 쿼리 대신 평균 벡터를 사용하는 방식으로도 다른 전통적인 방법처럼 일반적인 음원 분리에 사용될 수 있다는 것을 의미한다. 오디오 쿼리를 이용한 실험의 두 결과와 비교해보면 평균 벡터를 사용한 경우의 성능이 그 가운데에 위치해 있는 것을 볼 수 있다. 이는 본 연구에서 제안한 프레임 워크를 사용할 때 분리하고자 하는 신호와 유사한 오디오 쿼리를 사용하기 어려운 경우 또는 해당 신호 클래스의 일반적인 특

표 4.4: MUSDB18 데이터셋 SDR 점수

	Vocals	Drums	Bass	Other
STL2[21]	3.25	4.22	3.21	2.25
WK[27]	3.76	4.00	2.94	2.43
RGT1[15]	3.85	3.44	2.70	2.63
JY3[9]	5.74	4.66	3.67	3.40
UHL2[25]	5.93	5.92	5.03	**4.19**
TAK1[23]	**6.60**	**6.43**	**5.16**	4.15
Ours (RQ)	4.69	4.05	2.98	2.02
Ours (GQ)	5.48	4.59	3.45	3.26
Ours (mean)	4.90	4.34	3.09	3.16

성과 유사한 경우에 평균 벡터를 사용하는 방식으로 동작이 가능하고 어느정도의 성능을 보장해준다는 것으로 볼 수 있다.

또한, 제안 방법에는 성능 개선의 여지가 남아있다. 본 연구의 목적은 성능 향상이 아닌 음원 분리 연구의 확장성을 높이기 위함이기 때문에 음원 분리 연구에서 가장 일반적으로 사용되는 U-net을 음원 분리 네트워크로 사용했는데 U-net 이외의 다른 네트워크 구조를 사용하여 더 높은 성능을 기대할 수 있다. 이는 제안한 프레임워크가 U-net 구조에만 한정되지 않으며 다른 분리 네트워크 구조에도 적용될 수 있기 때문이다. 더불어 다중 채널 Wiener 필터(multi-channel Wiener filter)와 같은 후처리 기술의 사용도 가능하다.

제 5 장 결론

5.1 연구 의의

본 연구에서는 음원 분리의 확장성을 높이기 위해 음원 분리 과정에서 오디오 쿼리를 받아 보조 입력으로 활용하는 오디오 쿼리 기반 음원 분리라는 새로운 프레임워크를 제안했다. 제안된 프레임워크는 오디오 쿼리를 잠재 공간의 벡터로 인코딩하는 쿼리 인코딩 네트워크와 인코딩된 잠재 벡터를 이용해 합성 음원에서 쿼리와 유사한 소스를 분리해내는 음원 분리 네트워크로 구성되어 있다. 더불어 본 연구에서는 음원 분리 네트워크를 U-net 기반의 구조를 이용했지만 제안한 프레임워크는 U-net을 제외한 다양한 구조의 음원 분리 네트워크에도 적용이 가능하다.

제안한 기법이 오디오 쿼리 기반 음원 분리에 적합함을 보이기 위해 테스트셋의 샘플들을 잠재 벡터로 인코딩한 뒤 t-SNE를 이용하여 시각화한 실험에서는 쿼리 인코딩 네트워크가 처음 보는 샘플들에 대해서도 잠재 공간 상에 서로 다른 점으로 맵핑하면서 강인하게 동작하며 음악적 특성을 잘 잡아내는 것을 보였다.

또한 오디오 쿼리를 이용한 음원 분리 실험은 제안한 프레임워크가 오디오 쿼리 기반 음원 분리에 적합함을 보였으며 게다가 데이터셋에서 같은 클래스로 정의된 악기들 중 특정 악기만을 분리해내는 결과를 보여 zero shot 음원 분리의 가능성까지 보였다.

쿼리 인코딩 네트워크가 형성한 잠재 공간에 대한 분석은 다양한 실험을 통해 진행되었다. 음원 분리에 사용되는 잠재 벡터에 따른 성능 비교 실험과 세분화된 클래스 정보를 이용한 음원 분리 성능 비교 실험은 음원 분리 성능과 밀접한 관계

를 갖고 있음을 보여주었고 서로 다른 악기의 잠재 벡터를 보간하여 음원 분리를 수행했을 때 보간 비율에 따라 분리되는 악기의 비율이 달라지는 결과를 확인할 수 있었다. 음원 분리 결과가 연속적이며 잠재 벡터를 통한 조절이 가능하다는 것은 음악의 편곡과 같은 분야에서도 유용하게 쓰일 수 있어 제안한 프레임워크의 효용성을 더해준다.

앞서 서술한 잠재 공간에 대한 분석 실험과 더불어 t-SNE를 통해 이상치로 추정되는 샘플들이 해당 악기의 분포의 외각에 위치한 것을 통해 제안한 프레임 워크의 쿼리 인코딩 네트워크가 해석 가능한 잠재 공간으로 인코딩하는 것을 보였다.

MUSDB18 데이터셋에 대한 정량 평가 실험을 통해 이후의 오디오 쿼리 기반 음원 분리 연구의 정량적 평가 잣대로 쓰일 수 있는 기준을 제시하였으며 기존 연구들과의 비교 실험은 제안된 프레임워크가 오디오 쿼리 기반 음원 분리뿐만 아니라 일반적인 음원 분리 목적으로도 쓰일 수 있음을 뒷받침했다.

또한 본 연구에서는 제안한 프레임워크의 특성을 활용하여 음원 분리를 반복적으로 수행하여 잠재 벡터를 정제하는 과정을 보였다. 이 방식은 분리하려는 소스의 특성이 해당 악기의 일반적인 특성과 거리가 멀 때 도움이 되는 것을 실험을 통해 보였다.

5.2 향후 연구

본 연구에서 제안한 방식은 음원 분리 과정에서 클래스 정보 대신 주어진 오디오 쿼리의 내용만을 활용하기 때문에 여러 방면으로 응용이 가능하다. 후속 연구로써 음원 분리 연구를 비지도 학습 방식으로 접근하는데 활용이 가능할 것으로 보인다. 음원 분리 연구 데이터셋은 단일 소스를 모으기가 어렵다는 한계점으로 인해 크기가

작지만 음악 태깅, 악기 인식과 관련된 데이터셋[39, 40]은 단일 소스를 제공하지는 않아도 음원 분리 연구 데이터셋에 비해 크기가 압도적으로 크다는 장점이 있다. 본 연구에서 제안한 데이터 샘플링 방식은 학습 과정에서 여러 음원이 합쳐진 경우에도 분리가 가능하도록 만들고 오히려 더 좋은 방향으로 학습이 되도록 이끌었기 때문에 앞서 언급한 대량의 데이터셋을 음원 분리 연구에 활용할 수 있는 여지가 있다.

제안한 프레임워크는 분리하려는 신호를 지정하는 방법으로 적어도 같은 악기의 샘플을 요구하는 방식으로 동작한다. 이는 오디오 쿼리 기반 음원 분리에서 가장 직관적인 접근 방법일 수 있으나 사용자적인 측면에서는 이를 적극 활용하기 어려울 수도 있다. 따라서 제안한 프레임워크의 사용성을 더 높이는 측면으로도 후속 연구를 진행할 예정이다. 한 가지 예시로 어떠한 음악의 제목이나 관련된 정보를 검색하고 싶을 때 스마트폰 앱을 이용해 음악의 일부분을 들려주고 검색하는 방식도 있지만 목소리로 해당 곡의 멜로디를 허밍하는 방식으로도 검색이 가능하게 해주는 방식도 있다. 이는 쿼리로 주어진 허밍에서 음의 상대적 변화와 관련된 특징을 포착하여 검색에 이용하는 방식으로 동작한다. 이에 착안하여 네트워크 학습 시 쿼리로 주는 오디오를 전처리를 통해 바꾸거나 쿼리에서 상대적인 음 변화 등의 정보만을 추출하여 분리에 사용하는 방식으로 학습을 해볼 수 있다. 이를 위해 주어진 오디오에서 높낮이 또는 음색과 같이 원하는 정보만을 분리해내는 연구들의 방법을 적용해 볼 수 있다.

마지막으로 few-shot 음원 분리와 관련된 연구를 꼽을 수 있다. 본 연구의 실험에서 제안한 프레임워크가 같은 클래스에 정의된 악기들 중 특정 악기만을 분리해내는 결과를 보여 few-shot 음원 분리에 대한 가능성을 보여주었다. 저자의 아는 바에 의하면 음원 분리 연구들 중 이를 주제로 다룬 연구도 없다. 새로운 음악이 만들어지고 새로운 장르가 계속해서 생겨나듯이 악기도 계속해서 변화하기 때문에 few-show 음원 분리 연구는 의의가 크다고 여겨진다. 또한 진행된 연구들이 없기 때문에 few-

shot 음원 분리의 평가 기준도 모호하다. 따라서 어떻게 few-shot 또는 zero-shot 음원 분리에 접근할 것인지 외에 분리 결과에 대한 평가 기준에 대해서도 제대로된 평가가 이루어질 수 있도록 연구가 진행되어야 한다.

참고 문헌

[1] Sebastian Ewert and Meinard Müller, “Using score-informed constraints for nmf-based source separation,” *in 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 129–132.

[2] John R Hershey, Zhuo Chen, Jonathan Le Roux, and Shinji Watanabe, “Deep clustering: Discriminative embeddings for segmentation and separation,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp 31–35. IEEE,2016.

[3] Xun Huang and Serge Belongie, “Arbitrary style trans-fer in real-time with adaptive instance normalization,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp 1501–1510.

[4] Jean-Louis Durrieu and Jean-Philippe Thiran, “Musical audio source separation based on user-selected f0 track.” *In International Conference on Latent Variable Analysis and Signal Separation*, pp. 438–445.

[5] Andreas Jansson, Eric J. Humphrey, Nicola Montecchio, Rachel M. Bittner, Aparna Kumar, and Tillman Weyde, “Singing voice separation with deep u-net convolutional networks,” in *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR*, Suzhou, China, October 23-27, 2017, pp. 745–751.

[6] Tero Karras, Samuli Laine, and Timo Aila, “A style-based generator architecture for generative adversarial networks,” arXiv preprint arXiv:1812.04948

[7] Diederik P. Kingma and Max Welling, “Auto-encoding variational bayes,” in *2nd International Conference on Learning Representations, ICLR* , Banff, AB,Canada, April 14-16, 2014, Conference Track Proceedings.

[8] Ulyanov, Dmitry, Andrea Vedaldi, and Victor Lempitsky, “Instance normalization: The missing ingredient for fast stylization,” arXiv preprint arXiv:1607.08022, 2016.

[9] Jen-Yu Liu and Yi-Hsuan Yang, “Denoising auto-encoder with recurrent skip connections and residual regression for music source separation,” in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 773–778.

[10] Antoine Liutkus, Fabian-Robert Stöter, Zafar Rafii,Daichi Kitamura, Bertrand Rivet, Nobutaka Ito, Nobu-taka Ono, and Julie Fontecave, ”The 2016 signal separation evaluation campaign,” in *Proceedings of Latent Variable Analysis andSignal Separation - 12th International Conference,LVA/ICA,* Liberec, Czech Republic, August 25-28, 2015, pp. 323–332.

[11] Laurens van der Maaten and Geoffrey Hinton, “Visualizing data using t-sne,” *Journal of machine learning research*, vol. 9, pp.2579-2605, Nov 2008.

[12] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, 2016, pp. 483–499.

[13] Sungheon Park, Taehoon Kim, Kyogu Lee, and Nojun Kwak, “Music source separation using stacked hourglass networks,” in *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR* , Paris, France, September 23-27, 2018, pp. 289–296.

[14] Zafar Rafii, Antoine Liutkus, Fabian-Robert Stöter, Stylianos Ioannis Mimilakis, and Rachel Bittner. The MUSDB18 corpus for music separation, December2017.

[15] Gerard Roma, Owen Green, and Pierre Alexandre Tremblay, "Improving single-network single-channel separation of musical audio with convolutional layers," in *International Conference on Latent Variable Analysis and Signal Separation,* 2018, pp. 306–315.

[16] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, 2015, pp. 234–241.

[17] P. Seetharaman, G. Wichern, S. Venkataramani, and J. L. Roux, "Class-conditional embeddings for music source separation," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019,pp. 301–305.

[18] Hyeong-Seok Choi, Jang-Hyun Kim, Jaesung Huh, Adrian Kim, Jung-Woo Ha, Kyogu Lee, "Phase-aware Speech Enhancement with Deep Complex U-Net," arXiv preprint arXiv:1903.03107, 2019.

[19] Kingma, Diederik P., and Jimmy Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.

[20] O. Slizovskaia, L. Kim, G. Haro, and E. Gomez, "End-to-end sound source separation conditioned on instrument labels," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 306–310.

[21] Daniel Stoller, Sebastian Ewert, and Simon Dixon, "Wave-u-net: A multi-scale neural network for end-to-end audio source separation," in *Proceedings of the*

19th International Society for Music Information Retrieval Conference, ISMIR, Paris, France, September 23-27, pp. 334–340.

[22] Fabian-Robert Stöter, Antoine Liutkus, and NobutakaIto, "The 2018 signal separation evaluation campaign," in *International Conference on Latent Variable Analysis and Signal Separation*, 2018, pp. 293–305.

[23] Naoya Takahashi, Nabarun Goswami, and Yuki Mitsufuji, "Mmdenselstm: An efficient combination of convolutional and recurrent neural networks for audio source separation," in *2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC)*, 2018, pp. 106–110.

[24] Naoya Takahashi and Yuki Mitsufuji, "Multi-scale multi-band densenets for audio source separation," in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2017, pp. 21–25.

[25] Stefan Uhlich, Marcello Porcu, Franck Giron, Michael Enenkl, Thomas Kemp, Naoya Takahashi, and Yuki Mitsufuji, "Improving music source separation based on deep neural networks through data augmentation and network blending," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP)*, 2017, pp. 261–265.

[26] Jun Wang, Jie Chen, Dan Su, Lianwu Chen, Meng Yu,Yanmin Qian, and Dong Yu, "Deep extractor network for target speaker recovery from single channel speech mixtures," in *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association*, Hyderabad, India, 2-6 September 2018, pp. 307–311.

[27] Felix Weninger, John R Hershey, Jonathan Le Roux, and Björn Schuller, "Discriminatively trained recurrent neural networks for single-channel speech sepa-

ration," in *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2014, pp. 577–581.

[28] Tom White, "Sampling generative networks," arXiv preprint arXiv:1609.04468, 2016.

[29] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shecht-man, "Toward multimodal image-to-image translation," in *Advances in Neural Information Processing Systems,* 2017, pp. 465–476.

[30] Quan Wang, Hannah Muckenhirn, Kevin Wilson, Prashant Sridhar, Zelin Wu, John Hershey, Rif A. Saurous, Ron J. Weiss, Ye Jia, and Ignacio Lopez Moreno, "VoiceFilter: Targeted Voice Separation by Speaker-Conditioned Spectrogram Masking," arXiv preprint arXiv:1810.04826, 2018.

[31] Gabriel Meseguer-Brocal and Geoffroy Peeters. "Conditioned-U-Net: introducing a control mechanism in the U-Net for multiple source separations," 20th International Society for Music Information Retrieval Conference, Delft, The Netherlands, 2019.

[32] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville, "Film: Visual reasoning with a general conditioning layer," in *Proc. of AAAI (Conference on Artificial Intelligence)*, New Orleans, LA, USA, 2018.

[33] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets,"in *Advances in Neural Information Processing Systems,* 2014, pp. 2672-2680.

[34] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in*IEEE Inter-*

national Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017, 2017, pp. 2242–2251.

[35] Gino Brunner, Yuyi Wang, Roger Wattenhofer, and Sumu Zhao, "Symbolic music genre transfer with CycleGAN," *CoRR*, abs/1809.07575, 2018.

[36] Kaizhi Qian, Yang Zhang, Shiyu Chang, Xuesong Yang, and Mark Hasegawa-Johnson, "Zero-Shot Voice Style Transfer with Only Autoencoder Loss," arXiv preprint arXiv:1905.05879, 2019.

[37] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz, "mixup: Beyond empirical risk minimization," arXiv preprint arXiv:1710.09412 (2017).

[38] E. Manilow, G. Wichern, P. Seetharaman, and J. Le Roux, "Cutting music source separation some Slakh:a dataset to study the impact of training data quality and quantity," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, Oct. 2019

[39] Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson, "Fma: A dataset for music analysis," In *Proc. of International Society for Music Information Retrieval Conference (ISMIR)*, 2017.

[40] Eric Humphrey, Simon Durand, and Brian McFee, " Openmic-2018: an open dataset for multiple instrument recognition," In *Proc. of International Society for Music Information Retrieval Conference (ISMIR)*, 2018.

ABSTRACT

In recent years, music source separation has been one of the most intensively studied research areas in music information retrieval. Improvements in deep learning lead to a big progress in music source separation performance. However, most of the previous studies are restricted to separating a few limited number of sources, such as vocals, drums, bass, and other.

In this study, we propose a network for audio query-based music source separation that can explicitly encode the source information from a query signal regardless of the number and/or kind of target signals. The proposed method consists of a Query-net and a Separator: given a query and a mixture, the Query-net encodes the query into the latent space, and the Separator estimates masks conditioned by the latent vector, which is then applied to the mixture for separation. The Separator can also generate masks using the latent vector from the training samples, allowing separation in the absence of a query.

We evaluate our method on the MUSDB18 dataset and the Slakh dataset, and experimental results show that the proposed method can separate multiple sources with a single network. In addition, through further investigation of the latent space we demonstrate that our method can generate continuous outputs via latent vector interpolation.

주요어: 음원 분리, 오디오 쿼리
학번: 2018-22176